NAVAL POSTGRADUATE SCHOOL
Monterey, California

DISSERTATION

INERTIAL AND MAGNETIC TRACKING OF LIMB SEGMENT ORIENTATION FOR INSERTING HUMANS INTO SYNTHETIC ENVIRONMENTS

by

Eric Robert Bachmann

December 2000

Dissertation Supervisor: Michael J. Zyda

Approved for public release; distribution is unlimited.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY ***(Leave blank)***

2. REPORT DATE
December 2000

3. REPORT TYPE AND DATES COVERED
Ph.D. Dissertation

4. TITLE AND SUBTITLE
Inertial and Magnetic Tracking of Limb Segment Orientation for Inserting Humans into Synthetic Environments

5. FUNDING NUMBERS
ARO Proposal No.
40410-MA

N0003900WRDR053

6. AUTHOR(S)
Bachmann, Eric R.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
U.S. Army Research Office (ARO) Research Triangle Park, NC 27709-2211
U.S. Navy Modeling and Simulation Office (N6M) Washington, DC 20350-2000

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(maximum 200 words)*

Current motion tracking technologies fail to provide accurate wide area tracking of multiple users without interference and occlusion problems. This research proposes to overcome current limitations using nine-axis magnetic/angular rate/gravity (MARG) sensors combined with a quaternion-based complementary filter algorithm capable of continuously correcting for drift and following angular motion through all orientations without singularities.

Primarily, this research involves the development of a prototype tracking system to demonstrate the feasibility of MARG sensor body motion tracking. Mathematical analysis and computer simulation are used to validate the correctness of the complementary filter algorithm. The implemented human body model utilizes the world-coordinate reference frame orientation data provided in quaternion form by the complementary filter and orients each limb segment independently. Calibration of the model and the inertial sensors is accomplished using simple but effective algorithms. Physical experiments demonstrate the utility of the proposed system by tracking of human limbs in real-time using multiple MARG sensors.

The system is "sourceless" and does not suffer from range restrictions and interference problems. This new technology overcomes the limitations of motion tracking technologies currently in use. It has the potential to provide wide area tracking of multiple users in virtual environment and augmented reality applications.

14. SUBJECT TERMS
Micromachined Sensors, Complementary Filtering, Quaternions, Motion Tracking, Networked Virtual Environments, Quaternion/Vector Pairs, Human Modeling, MARG Sensors, Inertial Sensors, Magnetic Sensors

15. NUMBER OF PAGES
195

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT
Unclassified

18. SECURITY CLASSIFICATION OF THIS PAGE
Unclassified

19. SECURITY CLASSIFI- CATION OF ABSTRACT
Unclassified

20. LIMITATION OF ABSTRACT
UL

Approved for public release; distribution is unlimited

INERTIAL AND MAGNETIC ANGLE TRACKING OF LIMB SEGMENTS FOR INSERTING HUMANS INTO SYNTHETIC ENVIRONMENTS

by

Eric Robert Bachmann
B.A., University of Cincinnati, 1983
M.S., Naval Postgraduate School, 1995

Submitted in partial fulfillment of the
requirements for the degree of

DOCTOR OF PHILOSOPHY IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
December 2000

Author: Eric R Bachmann
Eric R. Bachmann

Approved by:
Michael J. Zyda, Professor of Computer Science

Robert B. McGhee
Robert B. McGhee
Professor of Computer Science

Xiaoping Yun,
Professor of Electrical Engineering

Man-Tak Shing, Associate
Professor of Computer Science

Donald P. Brutzman, Assistant
Professor of Applied Science

Approved by: Dan C Boger
Daniel Boger, Chair, Department of Computer Science

Approved by: Anthony Ciavarelli
Anthony Ciavarelli, Associate Provost of Instruction

ABSTRACT

Current motion tracking technologies fail to provide accurate wide area tracking of multiple users without interference and occlusion problems. These limitations make difficult the construction of a practical and intuitive interface, which allows humans to be inserted into networked virtual environments in a fully immersive manner. Advances in the field of miniature sensors make possible inertial/magnetic tracking of human body limb segment orientation without the limitations of current systems. Due to implementation challenges, inertial/magnetic sensors have not previously been used successfully for full body motion capture. This research proposes to overcome these challenges using multi-axis sensors combined with a quaternion-based complementary filter algorithm capable of continuously correcting for drift and following motion through all orientations without singularities.

Primarily, this research involves the development of a prototype tracking system to demonstrate the feasibility of hybrid RF/magnetic/inertial motion tracking. Construction of inertial/magnetic (MARG) sensors is completed using off-the-shelf components. Mathematical analysis and computer simulation are used to validate the correctness of the complementary filter algorithm. The implemented human body model utilizes the world-coordinate reference frame orientation data provided in quaternion form by the complementary filter and orients each limb segment independently. Calibration of the model and the inertial sensors is accomplished using simple but effective algorithms. Physical experiments demonstrate the utility of the proposed system. These experiments involve the tracking of human limbs in real-time using multiple inertial sensors.

The motion tracking system produced has an accuracy which is comparable and a latency which is superior to active electro-magnetic sensors. The system is "sourceless" and does not suffer from range restrictions and interference problems. With additional MARG sensor packages, the architecture produced will easily scale to full body tracking. This new technology overcomes the limitations of motion tracking technologies currently

in use. It will provide wide area tracking of multiple users in virtual environment and augmented reality applications.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

To achieve any worthwhile goal, all of us must stand on the shoulders of those who have come before and learn from our gifted contemporaries. Otherwise, we would all still be wearing animal skins and wondering what shape the wheel should have. Completion of this research was only possible due to the talent and patience of many special individuals. This dissertation serves as documentation of their contributions.

Most of all I would like to express my gratitude to Dr. Robert McGhee. From beginning to end, his patience as well as sheer genius made possible any progress that was made. His intellectual contributions to this work as well as myself are tremendous. However, it is for his role as my supporter and his efforts to boost my confidence when it waned that I am most thankful. It is a measure of my respect for Dr. McGhee, that even after all the time we have spent together, that I am still unable to call him by his first name.

During the period under which this work was completed, the advice and guidance of Michael Zyda did much to help me to limit the scope of the work undertaken and keep me focused on the 'real goal.' The real goal being of course to finish the dissertation and graduate. There were many extended periods during which I felt little or no progress was being made. I often waited for Mike to ask what the heck I was up to, but he was patient and simply waited for the results which he seemed to know would come even if I was not so sure.

This is a dissertation for a doctorate in Computer Science. However, it involved the study of several topics that are normally associated with Electrical Engineering. I am not an electrical engineer, but through many hours, the patience and teaching skills of Dr. Xiaoping Yun brought me as close as I will ever be. Even though Xiaoping is extremely brilliant, he was still always able to come down to my level to successfully teach me the complexities of random processes, the frequency domain and optimal filtering. The difficulty of this acheivement should not be underestimated.

In the beginning there was a qualifying exam... It was full of automata and algorithms. Mantak Shing generously gave me his time and lent me his knowledge to help me to successfully prepare for the exam. The mind of Don Brutzman is truly amazing. I look at something and I am only able to see it one way. Don looks at the same thing and is able to see it from an infinite number of viewpoints. I am very grateful for his multi-dimensional advice and guidance.

I have had the pleasure of advising numerous students who have contributed to this research. Among these, the contributions of Ildeniz Duman and Umit Usta were particularly significant. Funding was provided by the U.S. Army Research Office (ARO), the U.S. Navy Modeling and Simulation Office (N6M), and the Naval Postgraduate School. Without the support of these organizations this research could not have been completed.

A dissertation designed to demonstrate human body tracking using inertial/magnetic sensors would not be very creditable without any inertial/magnetic sensors. Though much of the time I had no idea what he was talking about, it was the skills and expertise of Doug McKinney that produced the first functioning MARG sensors.

The period during which this research was conducted was marked by tremendous change in my life. Much of it was extremely difficult and I often felt that everything was falling apart. Ann Flood helped me keep all the pieces together. Geographically distant, but close nonetheless, the support of my sisters Gail and Lynn and my mother also helped me to keep those pieces together. To all my friends who patiently listened to my whining, I also give my thanks.

Only my daughter, Carol, knows what it has been like to live me as I worked to finish this dissertation. She has put up with me and has been far more important to its completion than she will even know. I often think of her and wonder how one so young ever got to be so wise. Carol, I think the good times may finally be here.

DEDICATION

For Carol and all for Carol

In memory of Robert G. Bachmann

I. INTRODUCTION

A. MOTIVATION

Fully developed networked synthetic environments (SE) stand to revolutionize the fields of education, training, business, retailing and entertainment. They will fundamentally alter our societies and the way in which mankind views the world. In the educational field, synthetic environments will offer the ultimate in hands-on and visualization of difficult concepts. They will allow training to transpire in a place much like that in which the skills being practiced will be used without exposure to possible hazards and at less cost. In the workplace, employees will be able to work "side by side" even though they may be physically separated by hundreds or even thousands of miles. Using synthetic environments, corporations will obtain a safe, economical and efficient method of testing new concepts and systems. Retailers will create virtual department stores where consumers will be able to try out products to an unprecedented degree before actually buying them. Using synthetic environments, the entertainment industry will be able to create entire worlds in which customers will be able to experience thrills and live out entire fantasy lives. [Ref. 21.][Ref. 97.]

The power of the synthetic environment lies in its ability to immerse users in a different world. The more complete the immersion, the more effective the synthetic environment. For complete immersion, the user should sense and interact with the synthetic environment in the same manner in which interaction with the natural world takes place. Interaction in the natural world results from body motion. Information regarding the surrounding environment is obtained through the five senses. Changes in body posture and position directly affect what is seen, heard, felt and smelled. The parameters sensed in the environment are altered and manipulated by the actions of the body. Thus, in order for a user to interact with a synthetic environment in a natural way and have the synthetic environment present appropriate information to the senses, it is imperative that data regarding body motion and posture be obtained. Body posture and location data are also needed in multi-user environments to drive the animation of avatars which represent the actions of users of the environment to each other.

At this time, there is no practical and intuitive interface that allows an individual human to be inserted into a SE in a fully immersive manner. Numerous motion tracking technologies are currently in use, but each suffers from its own set of limitations. Depending on the technology, these limitations may include marginal accuracy, user encumbrance, restricted range, susceptibility to interference and noise, poor registration, occlusion difficulties and high latency. Due to these problems, real-time animations of avatars must be largely script-based using motion libraries. For the most part, only a single user may be tracked in a small working volume. Thus, none of the current technologies fulfills the need for wide-area tracking of multiple users. The ideal motion tracking technology must meet several requirements. It should have low latency, be tolerant to noise and other environmental interference, track multiple users and maintain both adequate accuracy and registration throughout a large working volume [Ref. 62.].

The primary reason current tracking systems fail to meet the requirements described above is the dependence of these systems on a generated "source" to determine orientation and location information. This source may be sent by transmitters to body-based receivers or it may be sent from body-based transmitters to receivers positioned at known locations throughout the working volume. Usually the effective range of this source is extremely limited or there may be compromises between resolution and range. Interference with or distortion of this source will at best result in erroneous orientation and position measurements. If the source is no longer received, it can cause a complete loss of track. Huge gains in reliability and capability would be achieved through the development of a "sourceless" sensor technology which could determine orientation and position without depending on an externally generated source.

The development of micromachined magnetometers and inertial sensors over the last few years makes it possible to determine orientation based on the passive measurement of physical quantities which are directly related to the motion and orientation of a rigid body to which they are attached. The "sourceless" nature of inertial and magnetic orientation tracking makes possible a full body posture tracking system that avoids the problems associated with current technologies and ultimately allows tracking over a virtually unlimited area.

Inertial/magnetic orientation tracking is based upon established algorithms in which local magnetic field, angular velocity, and linear acceleration data are combined to obtain estimates of location and orientation. It involves placing miniature sensor units on the body segments to be tracked. In the method discussed here, each unit contains a three-axis magnetometer, a three-axis angular rate sensor, and a three-axis accelerometer. In this document, nine-axes sensor units of this type are referred to as MARG (Magnetic, Angular Rate, Gravity) sensors.[Ref. 6.] Integration of angular rate sensor data provides the information necessary to calculate the orientation of a human body segment for relatively short time periods. However, sensor drift and bias errors associated with small and inexpensive sensors make it impractical to track orientation for long time periods. In the long term, accelerometers can be used to determine the direction of the local vertical by sensing acceleration due to gravity. In a similar manner, magnetometers can sense the direction of the local magnetic field. Thus, use of data from these complementary sensors can be used to eliminate drift by continuous correction of the orientation obtained using rate sensor data. In this manner a continuously accurate estimate of the orientation of each individual limb segment can be obtained.

The orientation obtained using MARG sensors is in an earth fixed coordinate reference frame. Using this information, each limb segment can be oriented within the synthetic environment without regard to the orientation of adjacent segments [Ref. 64.][Ref. 28.]. The posture of the user can then be reconstructed by simply attaching the representations of individual limb segments together in the same manner in which the corresponding segments on the body of the user are connected. There is no need for transformations between limb segment associated coordinate frames nor for determination of joint angles. Body posture is entirely determined based upon limb segment orientation and length.

It should be noted that though it is possible to determine limb segment orientation and hence body posture using only inertial and magnetic data, determining position requires double integration of linear acceleration data. The inherent noise, manufacturing defects, and measurement errors associated with low cost inertial sensors, and the quadratic growth of errors through double integration, makes uncorrected acceleration-based position tracking impractical for more than a very short period. Positioning of the user's avatar within the synthetic environment would thus be better accomplished through the use of a long range positioning system which is not susceptible to interference or noise to precisely locate a single body reference point. Depending on the accuracy required, GPS could be used in outdoor applications to provide the required position vector [Ref. 45.]. A more precise spread spectrum radio frequency (RF) positioning system could be used for indoor applications or applications requiring greater accuracy.[Ref. 24.]

In the complete inertial tracking system, individual MARG sensors will output the angular orientations of each tracked body segment. The outputs of these sensors will be conditioned and at least partially processed by a small wearable computer carried by the user. A position vector for at least one point on the body would be determined with the aid of an RF spread spectrum positioning system. These data would then be packaged into a serialized bit-stream and sent via wireless transmission to a base electronics package for further processing and submission to a synthetic environment. The resulting orientation and position data would be used to drive the animations of human avatars in a networked virtual environment and provide posture and location correct information to the senses of the user. If difficulties arise due to intermittent reception of RF positioning information, location can be estimated inertially for short periods of time.

B. GOALS

The research outlined in this document proposes to demonstrate the feasibility and capabilities of full body angle tracking by tracking human limb segments using multiple prototype MARG sensors.

1. Problem to be Solved

Several challenges have been overcome to bring a magnetic/inertial orientation tracking system to fruition. Inertial orientation tracking in high acceleration applications without serious drift error requires an integrated nine-axis sensor containing a three-axis accelerometer, a three-axis rate sensor and a three-axis magnetometer. Each sensor triad must be properly calibrated to determine sensor nulls and scale factors. Filtering and combining sensor data in a complementary manner requires the design of an efficient, but accurate software filter capable of tracking continuously in all orientations without singularities. Furthermore, using world-coordinate frame orientation data to drive the animation of an avatar requires development of a simplified human body model which allows independent positioning of each limb segment. Since the sensors can not be precisely mounted on each limb segment in a predefined position, the human model must take into account the offset between the body axes of each limb segment and the axes of the attached sensor. Finally, the animation of the avatar must take place with minimal lag and latency.

2. What is Fundamentally New

This research demonstrates a new technology for human body tracking in networked virtual environment applications. It shows that it is possible to construct a full-body tracking system capable of accurately determining body posture with minimal lag throughout a large working volume without occlusion problems. Unlike current body tracking technologies, the system is not continuously dependent upon any external source. This work describes the development of a novel nine-axis inertial sensor containing three orthogonal accelerometers, three orthogonal angular rate sensors, and three orthogonal magnetometers mounted in combination. At the core of the system is a complementary filter based upon quaternions. The software filter can track human body limb segments through all orientations without singularities. Drift corrections are performed continuously. Though the filter is nonlinear, it can be shown through nonlinear simulations and actual system performance that linear analysis of the filter is relevant and can by used as a method for selecting scale factors and for predicting performance. Animation of the avatar is accomplished using only orientation data. There is no need for complex kinematic computations to determine joint angles. Novel algorithms allow calibration of both the sensors and the human body model offsets quickly and accurately with no special equipment.

3. Contribution of this Research

This research demonstrates a new technology that overcomes the limitations of motion tracking technologies currently in use. The technology is capable of providing wide area tracking of multiple users for synthetic environment and augmented reality applications. This system makes a significant step toward "total immersion" of users in a networked synthetic environment by allowing them to interface with it using their natural bodies.

C. METHOD

Primarily, this research involves the development of a prototype MARG sensor tracking system including innovative calibration and angle tracking software. Examination of this implementation demonstrates the feasibility of a hybrid MARG/RF motion tracking system for networked synthetic environments.

Mathematical analysis, computer simulation and physical experiments are used to validate the correctness of the complementary filter algorithm as well as the human body model. The analysis is largely based upon linear approximation of the nonlinear problem. Frequency domain methods are used for analytic determination of system response characteristics. Nonlinear computer simulations are used to confirm the validity of the linear approximations.

Physical experiments have been completed to convincingly demonstrate the utility of the proposed system. These experiments involve the tracking of a human limb using prototype inertial sensors. Sensor data is provided to multiple quaternion filter software objects. Each quaternion orientation filter object corresponds to a particular human limb segment or segments and thus provides the orientation of it. These orientations are used to drive the animations of a human model in real-time.

Qualitative and quantitative results provide data for comparison to other motion tracking technologies. Preliminary attempts are made to estimate the performance parameters of the prototype system. System sensitivity to interference and noise is also examined.

D. DISSERTATION ORGANIZATION

This dissertation contains seven chapters.

- Chapter II presents a survey of motion tracking technologies currently in use with comments regarding the strengths and weaknesses of each. Included is a discussion of the performance parameters which are required to track the human body for real-time synthetic environment applications. Chapter II also provides a framework under which motion tracking technologies can be evaluated.

- Chapter III reviews different methods of representing the orientation of a rigid body with particular emphasis on quaternions and Euler angles. Various general methods of modeling the human body for synthetic environment applications are discussed as well.

- Chapter IV briefly presents the current state of micromachined sensor technology and reviews the fundamentals of software filter theory which pertain to human body tracking.
- Chapter V presents a description of a complementary filter based upon a quaternion representation of orientation. Analysis as well as simulation results for the complementary quaternion attitude filter are included.
- Chapter VI describes a prototype system for tracking human limb segments. The theory and algorithms used to calibrate the multi-axis sensors and the human body model are discussed.
- Chapter VII presents the results of experiments designed to quantify the performance of the prototype system. These data provide some indication of the performance which could be expected of a complete human body tracking system.
- The final chapter of this document presents conclusions and outlines the work which must be completed to build a complete human body tracking system capable of tracking multi-users in a large working environment.
- Appendix A contains detailed derivations of the Gauss-Newton iteration equations. Appendix B contains a derivation of the associated *X* matrix. Appendix C is a video demonstration of the body tracking system in operation.

II. SURVEY OF TRACKING TECHNOLOGIES

A. INTRODUCTION

The following survey is meant to establish the technological environment under which magnetic/inertial body tracking is introduced. Though specific examples of the various types of tracking systems are discussed, no attempt is made to comprehensively cover the multitude of tracking systems currently available on the market or being researched. Rather, the purpose is to establish the general limitations and performance capabilities of the various motion capture technologies available at the time of this writing.

B. MOTION TRACKING TECHNOLOGIES

In general, position and orientation tracking has seen insufficient innovation and development over the past decade. This continues to hamper advanced development of immersive systems that allow participants to enter and navigate simulated environments [Ref. 97.]. Today's commercial motion tracking systems are based on optical, magnetic and acoustic sources. Inertial sensing has been used for head tracking. RF positioning shows promise, but no small scale commercial systems are currently available for indoor use. The most popular trackers are active AC or DC magnetic systems. Before each of these technologies can be examined, two baselines must be established. First, in order to allow comparison of technologies, a "framework for suitability" is needed. Second, it is necessary to determine the specific performance characteristics that a human motion tracking system should have, based upon the dynamics of human body motion and research relating to human factors in synthetic environments.

a. Framework for Suitability

Several frameworks for use in the analysis of tracking technologies have been suggested [Ref. 62.], [Ref. 78.], [Ref. 21.]. Each proposes a similar method for categorizing the strengths and weaknesses of a particular technology. A basic framework which is based upon those mentioned above is provided here. Five key measures are proposed: *resolution*, *registration*, *responsiveness*, *robustness*, and *sociability*.

Resolution is the smallest change a system can detect. Poor resolution will allow the user to move without any corresponding change being expressed by the avatar within the synthetic environment. Without fine enough resolution, small details in the motion of a user will not be captured.

Registration is a measure of the correspondence between the position and orientation reported by the motion tracking system and the true position and orientation. Without adequate registration, it is not possible for individuals to interact with physical objects while immersed within the synthetic environment. Nor would it be possible for two users to physically interact with each other and perform simple actions such as a handshake. Registration is a function not only of tracking accuracy, but the also the fidelity of the correspondence between the avatar and the subject being tracked.

Overall responsiveness is determined by sample rate, data rate, and update rate. Responsiveness is fundamentally related to system *latency* or *lag*, which can be defined as the time delay between the movement of a tracked object and a corresponding update of the state of the synthetic environment. Lag which is imperceptible to the user will still degrade human performance due to dynamic registration errors. Systems with poor responsiveness make it difficult for the user to experience a feeling of presence. In some cases, latency can lead to simulator sickness. [Ref. 21.]

Robustness measures the susceptibility of a tracker to noise and interference within the operating environment. In a system with low tolerance to environmental noise, extreme errors may be present in the reported position and orientation. Inconsistency in these errors may make correction difficult using either software filters or lookup tables.

Sociability is an important measure of the suitability of a tracking system to wide area applications involving multiple users. Good sociability provides an extended range of operation under which resolution and registration are maintained as well a fitness for tracking multiple objects. There should be no collateral effects such as one remote object altering the reported position of another through either interference or occlusion.

b. Performance Requirements

To determine the minimum requirements for motion tracking performance, it is necessary to analyze the speed, force and frequency of human motion. Since hand and arm motions represent the quickest motions of the body, it can be assumed that a system capable of tracking the hands and arms will be able to track the rest of the body. Normal arm movements are accomplished with wrist tangential velocities of up to 3 m/s and accelerations not usually exceeding 5 to 6 g. Faster arm motions, such as throwing a baseball, may involve velocities of 37 m/s and accelerations in excess of 25 g. Normal hand motion bandwidth is around 2 Hz, while the fastest hand motions are in the 5-6 Hz range. Reflex actions may be on the order of 10 Hz [Ref. 12.]. Based on these values, a sampling rate on the order of 20 Hz would satisfy the requirements of the Nyquist sampling theorem [Ref. 14.]. In applications using sensors which are susceptible to noise, a general rule of thumb calls for 20 times oversampling. Thus, if 5 Hz is taken as the normal bandwidth of hand motions, human motion tracking requires a sampling rate of 100 Hz.

It is generally accepted that humans are more sensitive to changes in the rotation angle of proximal joints than in more distal joints. Changes in the position of a limb are usually experienced by the subject as a consequence of sensory receptors in the muscle propelling the motion [Ref. 40.]. The minimal passive changes humans will perceive in finger joints is about 2.5 degrees. For the wrist or elbow, a change of approximately 2 degrees is required. The minimal perceptible change in shoulder rotation is about 0.8 degrees [Ref. 21.]. Thus, a body sensor capable of resolving orientation to within 0.5 degrees should produce information which will not be in conflict with the kinesthetic nervous system of the user. Head tracking requires accuracy that is an order of magnitude greater than that required by body tracking applications. [Ref. 47.] Several authors call for orientation estimates which are accurate to within a few hundredths of a degree and position which is accurate to within a tenth of a millimeter [Ref. 21.][Ref. 29.]. Thus, any system which is able to meet the stringent requirements of head tracking would satisfy those of the rest of the body.

Humans are extremely sensitive to lag. Depending upon the task, time lags of greater than 100 msec can degrade performance. For head tracking applications, delays of as little as 60 msec between head motion and visual feedback are known to impair adaptation and may cause simulator sickness. If lag exceeds 300 msec, humans will begin to dissociate their movements from the displayed environment. A lag of greater than one second will force the user to adopt a move-and-wait strategy in order to complete a task. In general, as lag increases, user performance and speed decreases while the number of errors increases. [Ref. 21.][Ref. 47.]

In a typical SE system, there are multiple sources of lag. These include user input device lag, application-dependent processing lag, rendering lag, synchronization lag, and frame-rate-induced lag [Ref. 95.]. Often, it is difficult to determine that part of the total system delay which is due to the input device [Ref. 1.]. In any event, this lag should only account for a small portion of the total delay. Typical, input device lag ranges from 10 to 120 msec depending upon the type of filtering being performed and the mode of operation.

Kalman and Weiner predictive filtering can be used to extrapolate future time values based on previous user input data. To minimize the lag perceived by the user, the prediction algorithm normally attempts to project the user input data to the time at which results from these data reach the visual display. [Ref. 95.] This method reduces perceived lag as long as the user input device sampling rate is adequate and prediction too far into the future is not attempted.

What follows is a short survey of current methods used for motion tracking. Examples of some specific systems are provided to illustrate the current state of the art. Many of these systems have fairly high latency, marginal accuracy, moderate noise levels, and limited range. At this time, none is capable of fully meeting the need for a natural and intuitive whole body interface. Range restrictions produce a severe limitation in many of today's motion measurement technologies due to a dependence upon a generated source, which rapidly losses strength as range increases. Often, each user being tracked must compete with other users in the virtual environment either for access to the source or for space within a small working volume. This severely limits the number of users that can be tracked in the same area and essentially requires that all users be tracked in separate remote locations.

1. Mechanical Trackers

Mechanical tracking systems are perhaps the oldest motion tracking technology [Ref. 80.]. They provide the best means of providing haptic feedback to the user of a virtual environment. These systems are fairly accurate and have low latency. Current research generally involves using these tracking systems to calibrate other types of trackers. [Ref. 44.] Mechanical trackers can be placed in two separate categories. Here these categories will be termed body-based and ground-based.

Body-based systems utilize an exoskeleton which is entirely worn by the user of the synthetic environment. Goniometers within the skeleton linkages have a general correspondence to the joints of the user. These angle measuring devices provide joint angle data to kinematic algorithms which are used to determine end effector position as well as body posture. Since body-based systems are worn by the user, some other system must be used to ascertain position within the environment.

Attachment of the body-based linkages as well as the positioning of the goniometers present several problems. The soft tissue of the body allows the position of the linkages relative to the body to change as motion occurs. Even without these changes, alignment of the goniometer with body joints is difficult. This is especially true for multiple degree of freedom (DOF) joints. Since goniometers must be mounted externally, there will always be an offset from their centers of rotation to that of the actual joint. Human joints are not perfect hinges or spherical joints. Thus, any technology based upon this simplification will incur errors.

Due to variations in anthropometric measurements, body-based systems must be recalibrated for each user. This recalibration can be complicated and require an extensive period of time. Perhaps the most significant drawback of body-based systems is user encumbrance. Users must bear the weight of the exoskeleton as well as the annoyance of having an cumbersome framework attached to their body. The exoskeleton may make it difficult to interact with physical objects in a natural manner. For instance, it may be difficult to lie on the floor in certain positions since linkages may be between the user and the floor. All of these problems make it improbable that the user will become immersed within a synthetic environment and that a feeling of presence will be obtained.

Ground-based mechanical trackers typically have six degrees of freedom and provide the location and orientation of a single body segment. Thus, joint angle measurement error is not a factor. Typically, one end of a boom or shaft is either grasped by the user or attached to a device worn by user. The other end of the boom is attached to a fixed station by a 3 DOF joint. As the user moves the boom follows the motion. Encoders on the joint combined with the (possibly variable) length of the shaft provide the information needed to determine location and orientation within a synthetic environment. Ground-based mechanical tracking systems are limited to a range of approximately two meters by the inertia of the boom assembly. Longer shafts become too cumbersome and unwieldy.[Ref. 21.]

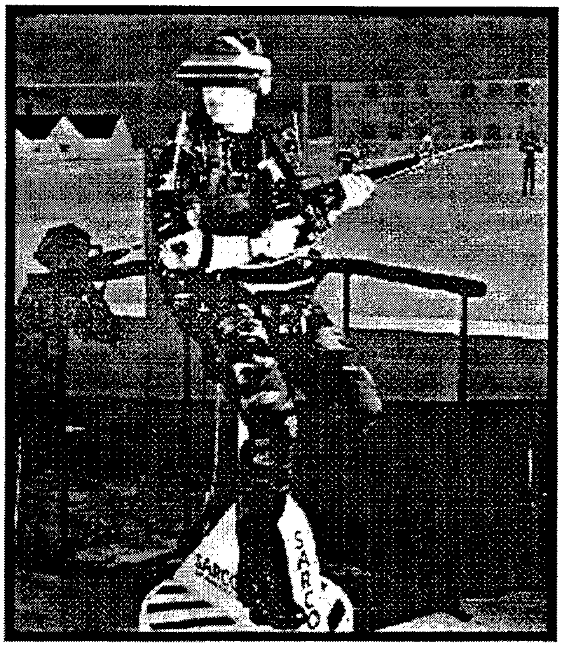


Figure 1: Exoskeleton tracking of the upper body

The BOOM (Binocular Omni-Orientation Monitor) is manufactured by FakeSpace Inc. It consists of a counterbalanced, 6 DOF shaft with a single immersive stereoscopic visualization display attached to one end. Shaft encoders produce translational and orientation accuracies of 0.16 inch and 0.1 degree respectively. Latency is on the order of 200 nsec. The operating radius is three feet horizontally with a vertical range of 2.5 feet. [Ref. 22.]

Mechanical motion tracking systems are fairly accurate and relatively inexpensive, but due to several limitations, they are unsuitable for accurately tracking multiple users in a large working volume. Body-based systems are difficult to calibrate and extremely cumbersome. In order to track over a large range, they must be combined with some other type of system. Ground-based systems, while highly accurate, can only track a single object over a very limited range.

2. Magnetic Trackers

Magnetic tracking using artificially generated sources is currently by far the most widely used technology for virtual and augmented motion tracking applications. For a relatively low cost, it can provide modest but reasonable accuracy with no serious obstruction or shadowing problems. These systems determine both position and orientation by using small sensors mounted on the body to sense a set of generated magnetic fields. The sensors contain three mutually perpendicular coils. As the coils are moved through the magnetic fields, the induced current within them will change. These changes in strength across the coils are proportional to the distance of each coil from the field emitter assembly. The emitter assembly itself is constructed of three mutually perpendicular coils that emit a magnetic field when a current is applied. Current is sent to these coils in a sequence that creates three mutually perpendicular fields during each measurement cycle. In all nine induced currents are generated within the sensor coils and used to calculate a position and orientation. Each of the three emitted fields creates one induced current in each of the three sensor coils, thereby allowing measurement of the nine elements of a rotation matrix associated with each sensor. [Ref. 73.]

At the time of this writing there are two primary manufacturers of magnetic tracking systems. The fundamental difference between their products is the type of current supplied to the emitter coils. Polhemus, Inc. uses alternating current (AC) to generate the field [Ref. 71.]. Ascension, Inc. utilizes direct current (DC). AC current creates continuously oscillating magnetic fields [Ref. 4.]. DC systems use pulsed magnetic fields and take measurements only after the fields have reached a steady state. This technique requires measurement of the ambient magnetic field so that it can be subtracted from the readings of the generated fields. Thus in addition to the nine measurements discussed above, three passive measurements of the constant magnetic field of the earth are required.

The shortcomings of magnetic tracking systems are directly related to the physical characteristics of magnetic fields. Magnetic fields decrease in power inversely with the square of the range as the distance from the generating source increases. This relationship limits these systems to a usable range which is no greater than the size of a small room. To simulate a larger working volume, user movement must be scaled or modified in some other manner [Ref. 65.] As emitter distance increases, position and orientation errors due to distortions of the generated field increase with the fourth power [Ref. 46.]. Thus, the accuracy of magnetic systems varies within the working volume. Distortions of the magnetic field come from several different sources. Changing magnetic fields produce eddy currents in metallic objects. The amplitudes of the eddy currents are proportional to the inverse cube of the transmitter to metal and receiver to metal separations. [Ref. 46.] The use of DC is an attempt to alleviate the eddy currents created by the continuously changing fields of AC systems. Ferromagnetic materials also produce magnetization fields due to their high permeability. These effects must be added to the distortions due to eddy currents. In addition to the eddy currents and possibly magnetization fields from metallic objects, magnetic sensors will also pick up noise from other magnetic fields that are generated within the environment by electrical devices. Such noise sources may include computer monitors, fluorescent lighting and any powered-up electrical wiring which is present within the surrounding walls. Even the wires connected to the receivers and transmitters themselves produce noise which may be significant [Ref. 46.].

In addition to software filtering, numerous techniques have been used in attempts to alleviate the error problems associated with magnetic tracking systems. Most are based upon the assumption that the magnetic characteristics within the working volume will not change. Function fitting has been attempted, but implementation requires a sacrifice of local accuracy to obtain better global accuracy [Ref. 44.]. In addition, a functional representation may not capture all of the details of the various distortions which may be present or may introduce variations of frequency which are higher than the frequencies of the error data. Lookup tables based on the reported position have met with only limited success. Livingston and State were able to reduce position errors by 79% within a two meter sphere surrounding the field transmitter. Construction of the table required a total 12,801 samples to obtain 720 valid table entries [Ref. 44.]. The same research determined that not only were orientation errors position dependent, but were orientation dependent as well. Since the look up table was based only upon position, correction of orientation errors was less successful. Creation of even a coarse lookup table which was dependent on orientation as well as position would have required taking over 332,826 samples within the two meter sphere [Ref. 44.].

Improvements in accuracy have also be made by varying the sampling frequency of the tracking system relative to the frequencies of the noise sources within the environment. Nixon et. al. reduced errors by sampling at twice the carrier frequency of the present electrical power and averaging of adjacent measurements [Ref. 46.]. However, when multiple noise sources operating at different frequencies were present, it was not possible to synchronize with all of them simultaneously.

Magnetic trackers are affected by many variables. Exact performance is difficult to quantify and is mostly application dependent. It has be shown to vary widely from the claims made by manufacturers. While manufacturers make latency claims on the order of 4 msec, observed delays on the order of 30 msec and may increase even further depending upon the number of sensors in use and the quality of filtering being performed [Ref. 1.]. Update rate also decreases with the number of sensors due to multiplexing. While

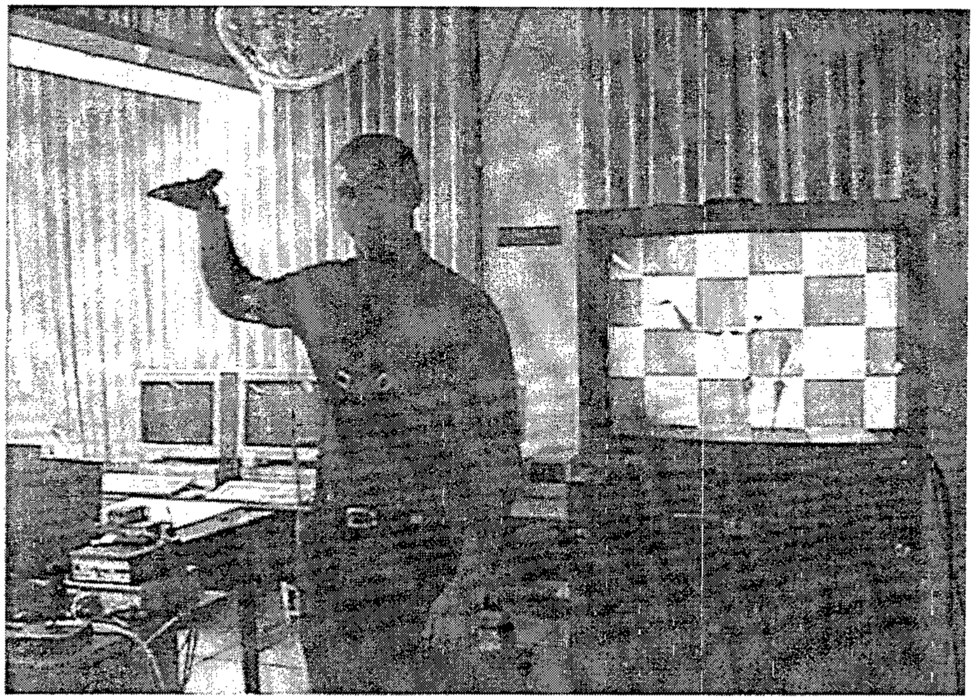


Figure 2: Electromagnetic Orientation Only Tracking of the Human Body From [Ref. 78.]

manufacturers promise operating ranges of 10 to 15 feet, in most cases the usable range has been shown to be much less [Ref. 46.]. Orientation accuracies of between 0.5 and 3.0 degrees are advertised. Position accuracies are given as anywhere from 0.3 to 3 inches. In both cases these values may vary widely depending upon the distance from the transmitter and the noise sources which are present.[Ref. 46.].

Skopowski did extensive work in tracking the upper body using electromagnetic motion trackers. His work included construction of a joint angle based kinematic model of the upper body. Difficulty in controlling figure motion indicated that the electromagnetic sensors used lacked sufficient position tracking accuracy. Therefore, the interface software used only orientation data for computing body joint angles. He concluded that the electromagnetic trackers lacked sufficient accuracy and registration to enable their use as a true six degree of freedom tracker in human body applications and called for the investigation of new tracking technologies to support the insertion of dismounted infantry into virtual environments [Ref. 78.].

The susceptibility of magnetic tracking systems to interference makes them unsuitable for robust synthetic reality applications. The presence of any magnetic materials or power sources within or near the working volume can severely degrade performance. Perhaps even more critical is the limited range of these devices. This limitation makes it nearly impossible to track more than one user in all but very specialized applications and restricts the size of the working volume to that of a small room.

3. Optical Sensing

Optical sensing encompasses a large and varying collection of technologies. More research is underway in this area than any other motion tracking technology. The cost and the performance of the different optical sensing technologies vary widely. Many are not capable of capturing motion data and processing it in real-time. The commonality between them is the dependence upon the sensing of some type of light. The light involved may or may not be visible to the eye. It may also be the focused light of a laser. It may be generated by a source under the control of the tracking system or it may be passive. Detectors may range from ordinary video cameras to lateral-effect diodes. In any case, optical systems suffer from occlusion problems whenever a required light path is blocked. Interference from other light sources may also be a problem. Lighting conditions must be controlled in order for the camera to consistently see objects in the environment. Depending upon the type of light in use, there may be severe range limitations.

Here, optical tracking systems are separated into three basic categories. *Pattern recognition systems* sense an artificial pattern of lights and use this information to determine position and/or orientation. *Image-based systems* determine position by using multiple cameras to track predetermined points on moving objects within a working volume. *Structured light and laser systems* have shown some promise, but little work appears to be under way to make this technology practical. None have been commercially marketed.

a. Pattern Recognition Systems

Pattern recognition systems can be outside-in or inside-out. In outside-in systems, the sensors (cameras in the case of optical systems) are fixed and the emitters are mobile. The sensors look into the working volume. Inside-out systems use sensors which are mounted on mobile objects and the emitters are fixed. These systems require elaborate preparation of the working volume. In both cases, position and orientation are calculated by viewing points of known intensity in known positions. Outside-in systems have a slight advantage in accuracy, since a small movement of the sensor will cause relatively large shifts in the apparent positions of emitters in view. The emitters themselves are usually infrared LEDs.

The HiBall tracker developed at UNC Chapel Hill is a classic outside-in system designed for head tracking. It utilizes a large number of ceiling-mounted infrared LEDs as emitters. The HiBall tracker or sensor is slightly larger than a golf ball. It contains six lenses and six photodiodes which are arranged so that each diode can view LEDs through several of the lenses. Position and orientation are determined by sequentially turning the LEDs in the ceiling on and off until it is determined which ones are in view of each of the photodiodes. Refinements over ten years of research and augmentation with inertial sensors has produced excellent performance. Position is accurate to 0.5 mm and orientation is resolved to within 0.02 degrees. The claimed update rate is greater than 2000 Hz with a latency of approximately 1 msec. [Ref. 93.]. The primary drawbacks of this implementation are its dependence on being under a specially prepared ceiling and its inability to track in all orientations. Current research aims at attempting to alleviate some of these problems and achieve passive optical sensing in a natural environment. [Ref. 93.]

The Honeywell LED array helmet tracker is a inside-out system designed for cockpit use. It uses an array with 4 LEDs mounted on a helmet. The LEDs are sequentially energized, and tracked by an infrared camera. The vector to each emitter is calculated using camera optical parameters and the known image of the source. From the four vectors, helmet orientation may be determined. [Ref. 25.]

b. Image Based Systems

Image based systems attempt to determine position through the use of multiple 2D images of the working volume. Stereometric techniques correlate common tracking points on the tracked objects in each image and use this information along with knowledge concerning the relationship between each of the images and camera parameters to calculate position. In some cases a single camera may be used and the process is based upon a sequence of images taken a different time instants. The tracking points are most often fiducial markers which are attached to the body being tracked. In order to calculate orientation, three noncollinear points on the each rigid body must be tracked. This process is prone to errors due to position inaccuracies, repeatability problems and non-simultaneous measurements [Ref. 66.]. The markers themselves may be either passive (retroreflective) or active (light-emitting diodes). A great deal of research effort is currently being expended on systems which are able to track natural objects in real-time without the add of markers.

All of these systems vary in the number of objects that can be tracked as well as the number of cameras that must be used. For many of the systems, the cost is quite high. In all cases there must be a compromise concerning lens focal length. A long focal length lens makes possible greater resolution over a smaller area than a short focal length lens. However, a long focal length has a smaller viewing area which will in turn reduce the size of the working volume unless additional cameras are added. Short focal lengths permit a larger working area, but at reduced accuracy. No matter how this compromise is resolved, these systems require that the entire working volume be within the view of several expensive cameras and thus even systems which might be capable of tracking natural objects will suffer from many of the limitations of sourced systems. All of these systems could be categorized as inside-out.

Passive marker measurement systems such as Vicon [Ref. 90.], HiRES 3D [Ref. 32.], and Peak Motus [Ref. 70.] use light sources placed very near each camera to generate light. This generated light is returned from the highly reflective markers. During the post-processing of the motion capture data, an operator of the system must assist with marker identification. The Peak Motus system has also been used without markers for analysis of athletic performances in a manual video acquisition mode. Higher end systems allow a frame rate of 60 Hz. Even for a small area, up to seven cameras may be required to achieve proper triangulation of the markers.

Though reflective marker research continues [Ref. 87.], in general, only active marker systems are currently able to produce the information necessary to drive an avatar in real-time. The advantage of the active marker approach is that the identity of each marker and thus the corresponding anatomical location is known immediately because the LEDs are sequentially pulsed by the control and data acquisition hardware. Data reduction is therefore greatly speeded up and a correspondence between multiple images can be found more quickly. “Phantom marker artifact” problems may be encountered due to the reflection of LED pulses from testing surfaces such as the floor. The range at which the LEDs may be detected is usually limited to less than eight meters.

Commercial active marker systems based on light-emitting diodes include Selspot II (Selspot Systems Ltd., Southfield, Michigan), OPTOTRAK (Northern Digital Inc., Waterloo, Ontario, Canada) and CODA (Charmwood Dynamics Ltd., Leicestershire, England). The Selspot II 3-dimensional motion measurement system allows the user to collect real-time 3D coordinates of up to 36 infrared LEDs attached to the test subject. It also calculates angle, acceleration, and moments. The CODA system is limited to providing stick figure animations and movement paths in a 2-D plane. Update rates for active marker systems are extremely high.

Reality fusion has released the GameCam system which is the first image based system motion capture system intended for use by the general public. In this system, the user must track their own location and position by viewing themselves on the screen. This low cost system uses a single standard PC camera to capture the motion of the user. Any motion information received from the camera that corresponds to an applicable portion of the screen image will alter the game environment. [Ref. 74.]

Camera tracking of natural objects without the aid of markers is considered by many researchers to be the final solution to human motion tracking challenges. It is largely based on computer vision techniques. It is felt that eliminating the need for fiducial markers will allow greater mobility and a deeper sense of immersion. To make this approach work it will still be necessary to position numerous expensive cameras throughout the working volume. A varied array of algorithms are being used to register objects in the video image or images captured by the camera with synthetic environment models. Some of the more common algorithms include mesh-based modeling, neurofuzzy classification, simple shape fitting, feature extraction based tracking and shape-volume approximation. Most of these algorithms are computationally demanding and are thus unable to deliver high quality motion capture data in real-time using current processing power. Often, several algorithms can be used in conjunction with one another. Mesh-based modeling breaks the video image into patches. The vertices of the patches can then be used as the nodes of a mesh. To register an object in the scene, a correspondence must be found between a given mesh model and the mesh which was created using the video image. Neurofuzzy classification uses a neural network which has been trained to recognize objects within the video image. Once an object has been recognized, knowledge of the camera parameters can be used to derive the position of the object. The basic premise of feature extraction and matching is that accurate 2D tracking of some basic distinctive features of an object in a sequence of images can lead to 3D tracking of the object. The most commonly used features are lines, points and curves. This technique is faster than more complicated methods, but is sensitive to image noise and occlusion. Simple shape fitting attempts to fit polyhedral, cylindrical or spherical models to candidate objects in the scene. Surface-volume approximation is similar to simple shape fitting and is usually combined with another technique such as mesh-based modeling.

c. *Structured Light and Laser Systems*

Structured light systems use lasers or beamed light to create a plane of light that is swept across the image. Some systems use a laser to scan points, the entire scene, or randomly to determine positions [Ref. 62.]. These systems sense the person, not just joints or points, thus, a person's body can be a virtual icon, rather than being artificially created from limited information. This is primarily a mapping technique, and is too slow for position tracking.

Laser Radar or *Ladar* measures the time of flight of laser light to an object and back. This gives distance information Three such measurements can be used to triangulate the position. If the angle of the laser beam is known, then only one measurement can give position. These systems are capable of providing very accurate distance information but resolution may be poor. Ladar is more appropriate for long distance measurements though the diffuse reflections may only have one sixth of the strength of the original beam. [Ref. 62.]

Laser Interferometers require retroreflectors or mirrors be attached to the tracked object. Laser light is directed to the reflector and the phase of the reflection is compared to the original light. An interference pattern is created and incremental distance information is found. Only incremental distance changes in distance are measured, so a position correction must be made to maintain registration. The correction may be found using laser radar. The orientations of the object be tracked are limited to those in which the reflector is accessible to the laser beam. This method is very accurate and precise. However, it is probably not suitable for measuring humans.

Structured light and laser systems are all susceptible to shadowing and occlusion problems. In general, they are quite complex and expensive. The measurement of orientation increases system complexity even further. They are more appropriate for mapping applications than dynamic tracking of human body motion.

4. Acoustic Trackers

Acoustic or ultrasonic trackers are an inexpensive alternative to magnetic trackers. They provide modest accuracies and update rates. Both outside-in and inside-out implementations are possible. Outside-in systems must divide the maximum update rate by the number of emitters being tracked. The number of tracked objects is not limited in inside-out systems.

The physics of sound limit the accuracy, update rate and range of acoustic tracking systems. Ranges are longer than that of magnetic trackers and magnetic interference is not a problem. However, a clear line of sight must be maintained. Thus, obstruction and shadowing can present difficulties [Ref. 21.]. Latency varies with distance due to the relatively slow speed of sound. Most current systems utilize 40 kHz tone pulses. Sound in this frequency band can be severely affected by noise from metallic objects such as jingling keys. Shorter wavelengths more accurately resolve distances, but quickly attenuate. In addition, high frequency omnidirectional radiators are expensive to implement and require more power.

Ultrasonic tracking systems can determine position through either time-of-flight and triangulation or phase-coherence. Phase-coherence trackers determine distance by measuring the difference in phase of a reference signal and an emitted signal detected by sensors. This difference is used to calculate changes in positions. Since this is an incremental motion technique, initial location must be determined by some other means and drift may be a problem. One of the major advantages of phase-coherence systems is higher data rates which allow filtering. Both types of systems can be adversely affected by echoes and reflections of sound waves.

5. Inertial and Magnetic Tracking

Though it is based upon well established algorithms, inertial and magnetic (MARG sensor) tracking is a relative newcomer to the motion tracking arena. It has been used to determine head orientation in virtual and augmented reality applications, but it has not yet found use in full body tracking applications. Inertial sensing is also finding expanded usefulness as a method of augmenting other motion tracking technologies. Practical inertial tracking is made possible by advances in miniaturized and micromachined sensor technologies, particularly in silicon accelerometers and rate sensors. These advances have been driven by the rapidly developing market for low cost automotive vehicle navigation and control systems. Unlike other sensor technologies, there is no inherent latency associated with inertial sensing. All delays are due to data transmission and processing. Thus, an orientation that is calculated using inertial sensor data is likely to be extremely accurate and have very low latency.

A naive approach to inertial orientation tracking would simply involve a single integration of angular rate data to determine orientation. However, this solution, which is found using only one type of sensor, would be prone to drift over time due to the buildup of small bias and drift errors. In order to avoid drift, inertial tracking systems make use of other complementary sensors to continuously correct the orientation estimate. Commonly, these sensors include an inclinometer or accelerometers to sense the vertical and a set of magnetometers to sense the direction of the local magnetic field. In order to track all orientations, there must be a separate accelerometer, rate sensor and magnetometer for each of the three coordinate axes of a rigid-body.

Theoretically, it is possible to determine position as well as orientation using inertial sensors. This is done on a daily basis by the inertial navigation systems of submarines and other platforms which must navigate without the aid of outside references. This dead reckoning performance is made possible through the use of very expensive and large sensors. Such dead reckoning is not possible with low grade inertial sensors for anything longer than relatively short time periods [Ref. 7.][Ref. 26.]. Without outside reference, position estimates based on inexpensive sensors will drift in a manner similar to orientation estimates based only on angular rate sensors.

Motion tracking of a two-joint, two-axis arm model using accelerometers and miniature gyroscopes was demonstrated Sakaguchi et al. [Ref. 77.] This research attempted to take advantage on the strengths of both sensor types and compensate for their weaknesses in a complementary manner. The method proposed does not use Euler integration of angular velocity data or complementary filter algorithms. Rotational and centrifugal accelerations are calculated based upon the kinematic relationships between the sensors and the links and the links with each other. The model is basically two-dimensional and no provisions are made to compensate for drift in the yaw axis. “Fingertip” position stated accuracy is 0.061mm.

Fuchs presented the first inertial system for head tracking applications [Ref. 29.]. This system utilized a fluid pendulum and three solid state piezsoelectric angular rate sensors. The initial system did not include a compass or magnetometers and thus drifted about the vertical axis. Subsequent systems include three orthogonal solid-state rate gyros, a two-axis fluid inclinometer and a two-axis fluxgate compass [Ref. 27.]. Intersense, Inc. was started as a result of this research and continues to produce inertial tracking devices designed for head tracking applications. Most the systems currently marketed are hybrids which use ultrasonic range-finding to determine or correct position. Advertised performance of the IS600 includes an angular accuracy of 0.25 degrees, translational accuracy of 0.25 inches and an update rate of up to 150 Hz. Though the response and accuracy of the systems is excellent, the use of Euler angles to internally represent orientation makes possible singularities in some orientations [Ref. 27.]. Sensor data is processed by a complementary separate-bias Kalman filter which requires periods of “still time” to correct for rate sensor drift. [Ref. 27.]. For most normal head tracking applications this is not a problem. However, in high acceleration applications requiring orientation tracking in all attitudes such limitations are not desirable. While InterSense is alleged to be developing and marketing a full body tracking system, the author is unaware of any research literature documenting such a system.

Henault researched software necessary to support inertial sensors capable of tracking all orientations. His work included the development of a quaternion attitude filter. The filter was tested with a computer simulated inertial tracker, [Ref. 35.]. Use of quaternions in the filter allowed objects to be tracked in all orientations and avoids the singularities associated with Euler angle based filters. Another important feature of the filter is reduced computational complexity since the incorporated filter uses no trigonometric functions.

6. RF Positioning

RF positioning has yet to be applied to the body tracking problem. Radio Frequency (RF) position systems are very fast and long range by their nature. Such systems have been developed for ships, planes, missiles and various civilian applications such as Long Range Navigation (Loran) and the Global Positioning System (GPS) [Ref. 45.]. All of these systems are designed to be used at extreme distances. In the past, they could only be used in such large-scale applications due to system errors in signal processing, [Ref. 13.]. Recent advances in RF systems technology however, make possible translational three degree of freedom tracking accuracy of a few millimeters at ranges of up to 100 meters, [Ref. 24.].

The speed of a radio signal is 2.99792458 x 10^8 meters per second. When a signal is transmitted, it takes a finite amount of time to travel from point *x* to point *y*. If the receiver knows the exact time the signal is transmitted and received, it can determine the amount of time the radio signal took to travel. Thus delta-time multiplied by the speed of the radio signal equals the range between the two points in meters. Using this method, a receiver-equipped object can determine its position through triangulation based upon its distance from several transmitters with known locations. This is the method used by GPS, [Ref. 45.]. Notice in this method that the receiver must accurately measure the exact times of transmission and reception with adequate temporal resolution. Alternatively, the difference in the time of arrival of a single signal at several locations can also be used to triangulate the position of an object equipped with a transmitter, [Ref. 24.]. This method does not require exact time synchronization between the receiver and transmitters.

Radio frequency devices have unique characteristics with both advantages and disadvantages. Some advantages important to position tracking are that radio frequencies can penetrate nonmetallic objects, such as walls and the human body. They are resistant to masking (hiding), easy to construct, and are scalable to both large and small areas. In a closed environment however, radio frequency device performance can be degraded due to reflections off surfaces, both metallic and nonmetallic, and is subject to attenuation when passing through objects, [Ref. 13.].

Spread-spectrum RF signals exhibit two characteristics important to positioning in a virtual environment. The first is excellent ranging ability, which allows accurate measurement of distance between two points, based on the phase difference in the pseudo noise (PN) code sequences of the transmitter and receiver. The second is code division multiplexing (CDM) which allows multiple transmitter-receiver pairs to compatibly share the same frequency at the same time, [Ref. 13.].

RF position tracking can be scaled to an area of any size. The accuracy of the system would be dependent upon the frequency, coding and signal processing implementation rather than the size of the area. A minimal system for 3 DOF tracking in a VE would require four stations placed at known locations within the area in which the tracking is to take place and a unit attached to the body to be tracked. The fixed location stations could be transmitters and the tracked unit a receiver as with GPS, or the roles could be reversed as described in [Ref. 24.]. In the former configuration, processors on the object itself could estimate the position of the object. In the latter configuration, position calculations would be made by a central processor in communication with each of the fixed location stations. Either configuration would be capable of producing highly accurate location data.

7. Hybrid Tracking Systems

Each type of tracking technology has its own set of strengths and weaknesses. The ultimate future of motion tracking almost certainly lies in hybrid systems. Many systems use one type of technology for sensing orientation changes and another for sensing position. Some merely use two separate technologies and choose whatever estimate seems to be the most accurate at a particular time instant. The best systems take data from multiple sensor types and use filtering algorithms to combine them and arrive at some type of optimal estimate. The inability of micromachined inertial sensor systems to calculate position for any extensive period of time in practical applications necessitates that any 6 DOF inertial system be a hybrid. The fact that inertial data lends itself to prediction through the use of motion derivatives has resulted in the use of inertial sensors in numerous efforts to combat latency problems.

In [Ref. 5.], Azuma demonstrates that predicting future head location using three rate gyroscopes and three linear accelerometers is an effective approach for significantly reducing dynamic errors in an augmented reality head tracking system. In this study, prediction caused dynamic accuracy to increase by factors of 5 to 10. Linear Kalman filters are used to estimate and predict translation terms and an Extended Kalman Filter (EKF) is used to estimate and predict orientation terms. Welch continued predictive work at UNC in [Ref. 94.], using a single-constraint-at-a-time (SCAAT) Kalman filter. Though a quaternion representation of orientation is used in the UNC research described above, in each case the orientation is converted to an Euler angle representation.

[Ref. 26.] describes a hybrid outside-in inertial/acoustic system called the *constellation.* This system uses an inertial navigation system which is aided by ultrasonic time-of-flight range measurements. The inertial subsystem determines position through double integration of triaxial accelerometer data. The ultrasonic ranging system uses a "constellation" of ceiling mounted acoustic beacons in a manner very similar to the optical HiBall head tracking system developed at the University of North Carolina, Chapel Hill. However, the ultrasonic system only calculates position. It does not calculate orientation. The stated reasons for using acoustic sensing as opposed to optical are cost, weight, and complexity. As with the UNC system, an extended Kalman filter is used to combine all sensor data and calculate an optimal position and orientation estimate. Acoustic range measurements are also individually processed using a SCAAT Kalman filter.

8. Other Technologies

Suryanarayanan and Reddy investigate the use of surface electromyographic (EMG) signals for tracking human movements for virtual environment application and control of telemanipulators [Ref. 72.]. This study concentrates on determining elbow-joint flexion and extension and using this information to drive an anthropomorphic telemanipulator. Accuracy is based on comparisons between the actual elbow joint angle and the angle produced by the telemanipulator. Use of EMG is difficult due to the nature of the EMG signals and large variations from one user to another. Signals vary based on both the speed of motion and the angle of motion. Limb loading and the plane of motion relative to the down vector will also affect the EMG signal. [Ref. 72.] utilizes a nonlinear, adaptive, intelligent system to track human arm movements. The system attempts to use an artificial neural network with fuzzy logic to compute an adaptive gain that compensates for the variation in the EMG signals due to speed of flexion. Only data from the biceps was used to compute joint angle. RMS joint angles errors where less than 20% during testing. Computation delays exceeded 150 msec.

Several technologies have produced good results in hand tracking applications. Most gloves combine a single 3D tracker to track hand position and orientation and multiple joint sensors for finger position. The *Dataglove* by VPL Research, measures bending in the proximal joints based on the attenuation of a light signal in each of two fiber optic strands sewn into the glove along the fingers and thumb. Sampling rate is 30 or 60 Hz [Ref. 67.]. The *Cyberglove* by Virtual Technologies includes either 18 or 22 resistive-strip sensors for finger bend and abduction, and thumb rotation [Ref. 67.]. Unlike the Dataglove, the mapping between the sensors and finger positions is linear. The strip sensors are more natural and comfortable to wear. Both of these glove technologies could be extended to the entire body through the use of a body suit. However, calibration for different users would most likely be difficult.

C. SUMMARY

This chapter presents a brief survey of technologies which are currently being researched or commercially marketed. The general limitations and performance capabilities of the various motion capture technologies are examined. None is capable of fully meeting the need for a natural and intuitive interface. In general, limited range, shadowing problems and susceptibility to interference make currently available systems unfit for tracking multiple users in the same work space. In addition, most sourced tracking systems fall short in categories of robustness and sociability.

The ideal tracking system would receive high marks in all measurement categories. It should be capable of accurately tracking multiple users in a large working volume with minimal lag. There should be few errors due to noise sources within the working volume or due to collateral effects associated with the tracking of multiple objects or users. The update rate should be adequate to capture the entire range of human motion. The ideal tracker should be not only be untethered, but also unobtrusive.

III. REPRESENTATION OF HUMAN BODY MOTION AND MODELING

A. INTRODUCTION

The human body can be modeled as a set of links or limb segments arranged in a tree-like structure. Individual limb segments can be treated as rigid bodies. Specifying the posture of the model involves descriptions of the orientation and position of the individual segments. Specification of the nature of the relationships between the links in the structure will determine whether the positions and orientations of the segments are described individually or are specified relative to one another. The formalisms chosen will have a bearing upon the ability to represent all orientations, computational efficiently, storage requirements, and transmission bandwidth requirements when operating within a networked synthetic environment architecture. In human body tracking applications, the type and quality of sensor input being used to drive the animation of the human model should also be considered.

This chapter discusses alternative methods of representing the orientation of the individual links of a human model and relating the links to one another. Possible methods of representing orientation considered include joint angles, Euler angles, and quaternions. Both simple and complex link relations and structures are examined and compared. Kinematic structures based upon homogenous transform matrices and quaternion/vector pairs are also examined.

B. RIGID BODY ORIENTATION REPRESENTATION

A "rigid body" is an idealization of a body with volume and mass which has a shape that cannot be changed. That is, such bodies are solid and completely inelastic. Numerous methods are available for expressing the orientation of a rigid body. Two of the more common methods are Euler angles and quaternions. Other methods of representation include direction cosines and vector-angle pairs. Direction cosines represent an orientation using the cosines of the angles an appropriate vector makes with the standard orthonormal basis for three space. The nine direction cosines associated with three unit vectors correspond to the nine elements of a rotation matrix. Vector-angle pairs specify an a vector and an angle of rotation about that vector. These are similar to a unit quaternion. Each method has its own set of advantages and disadvantages. Only Euler angle and quaternion representations are discussed here. More detailed discussions can be found in [Ref. 17.] and [Ref. 42.]

In order to represent the orientation of a rigid body, it is conventional to choose a coordinate system attached to an appropriate inertial frame, and then express all vectors in component form relative to these coordinates. A commonly used coordinate system is the local "flat Earth" system with an arbitrarily selected origin on the surface of the Earth with coordinate axes x, y, and z directed in the local north, east and down directions respectively. To specify orientation, it is also necessary, for each rigid body, to specify a "body fixed" coordinate system or frame which is attached to the rigid body. This is also an xyz system with x conventionally "out the nose," y "out the right side," and z down or "out the belly." (The reader may find it helpful to visualize an aircraft with positive axes pointing out the nose, right wing and bottom of the fuselage.) The superscript or subscript "E" is most often used to designate Earth coordinates, while "B" is typically used to signify body coordinates. The description of the orientation of a rigid body expresses the relationship between these two coordinate systems.

1. Euler Angles

Euler angles represent the orientation of a rigid-body using three rotations about specified axes. The axes may be orthogonal body fixed, orthogonal earth-fixed, or gimbal axes. Thus, when using Euler Angles, it is important that agreement be reached regarding the type of axes as well as the ordering of the rotations. If the order of rotations is first about a north axis, then about an east axis, and finally about a down axis, the associated angles are denoted by the reserved words "roll," "elevation," and "azimuth" respectively. When using the above set of “Euler” angles, there are also reserved symbols for each angle; namely, roll is designated by φ, elevation by θ, and azimuth by ψ. [Ref. 52.]

If the temporal order of rotations is reversed, body-axis rotations yield exactly the same orientation as Earth axis rotations. Specifically, starting with a given body in its reference orientation, if it is first rotated through the azimuth angle about its belly axis, then through elevation about its right side axis, and finally through the roll angle about its nose axis, the final orientation of the body will be exactly the same as if these rotations had been performed in the reverse order about the north, east and down axes of an earth fixed coordinate frame. [Ref. 52.]

Gimbal axes provide another way of defining Euler angles which helps to resolve the apparent temporal conflict. This approach is derived from the terminology and practice of naval gunnery and field artillery. To aim an artillery piece, it is necessary to tilt the gun barrel upward through an “elevation” angle so that a projectile will travel the desired distance when the gun is fired. It is also necessary to rotate the gun carriage to a proper “azimuth” angle so it points toward the target. Finally, in most modern guns, when the projectile is fired, the “rifling” in the gun tube imparts a “roll rate” (or “spin”) to the projectile to stabilize its flight toward a target. If the azimuth, elevation, and roll axes all intersect in a common point, then the mechanism that moves the gun is called a “gimbal” system. Thus, gimbal systems provide a mechanical means for achieving rotations. In this case the “temporal” order of the rotations does not matter. That is, the gun is “aimed” at the same point regardless of what temporal order the rotations are applied.[Ref. 51.]

a. Euler Angle Rotation

The position of a point in space can be described using a three dimensional point vector. If a rigid body is described in terms of point vectors, it can be rotated or oriented by rotating each vector individually. This may be completed by multiplying an appropriate rotation matrix times the point vectors. The rotated coordinates, $v' = [x'\ y'\ z']^T$, of a vector $v = [x\ y\ z]^T$ by an angle φ about the x axis is described by [Ref. 17.]

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\varphi & -\sin\varphi \\ 0 & \sin\varphi & \cos\varphi \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = [rot(x, \varphi)] \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{3.1}$$

Likewise, rotations of θ (elevation) and ψ (azimuth) about the y and z axes respectively can be accomplished by means of the following multiplications.

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = [rot(y, \theta)] \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{3.2}$$

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = [rot(z, \psi)] \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{3.3}$$

Thus, the relationship between the earth fixed coordinate system and the body fixed coordinate system can be expressed as a single rotation matrix R.

$${}^{E}v = [rot(z, \psi)][rot(y, \theta)][rot(x, \varphi)]\,{}^{B}v \tag{3.4}$$

$$= \begin{bmatrix} \cos\psi\cos\theta & \cos\psi\sin\theta\sin\varphi - \sin\psi\cos\varphi & \cos\psi\sin\theta\cos\varphi + \sin\psi\sin\varphi \\ \sin\psi\cos\theta & \cos\psi\cos\varphi + \sin\psi\sin\theta\sin\varphi & \cos\psi\sin\varphi + \sin\psi\sin\theta\cos\varphi \\ -\sin\theta & \cos\theta\sin\varphi & \cos\theta\cos\varphi \end{bmatrix} {}^{B}v = R\,{}^{B}v \tag{3.5}$$

It should be noted that this relationship applies regardless of the physical means by which the Euler angle rotations have been achieved. Rotation of the point vector v in Eq. (3.5) requires nine scalar multiplications and six additions. There are six trigometric functions.

b. Transforming Body Rates To Euler Rates

Unlike linear velocities which may be integrated to obtain position, the body rates p, q, and r about the body x, y, and z axes cannot be integrated to obtain Euler angles. That is

$$(\varphi, \theta, \psi) \neq \int (p \quad q \quad r)dt \tag{3.6}$$

This will be proved in the following paragraphs.

The angular rate of a rigid body in earth coordinates, ${}^{E}\omega$, is given by

$$ {}^{E}\omega = \begin{bmatrix}\omega_x\\ \omega_y\\ \omega_z\end{bmatrix} = \begin{bmatrix}0\\ 0\\ \dot\psi\end{bmatrix} + [rot(z,\psi)]\begin{bmatrix}0\\ \dot\theta\\ 0\end{bmatrix} + [rot(z,\psi)][rot(y,\theta)]\begin{bmatrix}\dot\varphi\\ 0\\ 0\end{bmatrix} \qquad (3.7) $$

where $\dot\varphi$, $\dot\theta$, and $\dot\psi$ are Euler rates measured about roll, elevation and azimuth Euler axes respectively. Eq. (3.8) below expresses angular rates about body-fixed axes in terms of angular rates about earth-fixed axes.

$$ {}^{B}\omega = \begin{bmatrix}p\\ q\\ r\end{bmatrix} = R^{-1}\ {}^{E}\omega = R^{T}\ {}^{E}\omega \qquad (3.8) $$

By the inverse law of transposed matrices and substitution of Eq. (3.7) into Eq. (3.8).

$$ {}^{B}\omega = [rot(x,\varphi)]^{T}[rot(y,\theta)]^{T}[rot(z,\psi)]^{T}\ {}^{E}\omega = [rot(x,\varphi)]^{-1}[rot(y,\theta)]^{-1}[rot(z,\psi)]^{-1}\ {}^{E}\omega \qquad (3.9) $$

$$ = [rot(x,\varphi)]^{T}[rot(y,\theta)]^{T}[rot(z,\psi)]^{T}\begin{bmatrix}0\\ 0\\ \dot\psi\end{bmatrix} + [rot(x,\varphi)]^{T}[rot(y,\theta)]^{T}\begin{bmatrix}0\\ \dot\theta\\ 0\end{bmatrix} + [rot(x,\varphi)]^{T}\begin{bmatrix}\dot\varphi\\ 0\\ 0\end{bmatrix} \qquad (3.10) $$

From the first term of Eq. (3.10), the rotational rate about an earth fixed down axis in body coordinates is given by

$$ \overrightarrow{{}^{B}\dot\psi} = [rot(x,\varphi)]^{T}[rot(y,\theta)]^{T}[rot(z,\psi)]^{T}\begin{bmatrix}0\\ 0\\ \dot\psi\end{bmatrix} = R^{T}\begin{bmatrix}0\\ 0\\ \dot\psi\end{bmatrix} = \dot\psi\begin{bmatrix}-\sin\theta\\ \sin\varphi\cos\theta\\ \cos\varphi\cos\theta\end{bmatrix} \qquad (3.11) $$

In a similar manner, the following are obtained from the second and third terms of Eq. (3.10) respectively.

$$ \overrightarrow{{}^{B}\dot\theta} = [rot(x,\varphi)]^{T}[rot(y,\theta)]^{T}\begin{bmatrix}0\\ \dot\theta\\ 0\end{bmatrix} = \dot\theta\begin{bmatrix}0\\ \cos\varphi\\ -\sin\varphi\end{bmatrix} \qquad (3.12) $$

$$ \overrightarrow{{}^{B}\dot\phi} = [rot(x,\varphi)]^{T}\begin{bmatrix}\dot\varphi\\ 0\\ 0\end{bmatrix} = \dot\varphi\begin{bmatrix}1\\ 0\\ 0\end{bmatrix} \qquad (3.13) $$

To obtain expressions of body rates in terms of Euler Rates and angles, Eq. (3.11), Eq. (3.10) and Eq. (3.11) are combined to produce

$$p = \dot{\varphi} - \dot{\psi}\sin\theta \tag{3.14}$$

$$q = \dot{\theta}\cos\varphi + \dot{\psi}\sin\varphi\cos\theta \tag{3.15}$$

$$r = -\dot{\theta}\sin\varphi + \dot{\psi}\cos\varphi\cos\theta \tag{3.16}$$

In order to solve for $\dot{\varphi}$, $\dot{\theta}$, and $\dot{\psi}$ in terms of p, q, and r, it should be noted that Eq. (3.15) and Eq. (3.16) involve only $\dot{\theta}$, and $\dot{\psi}$. Thus, multiplying Eq. (3.15) by *cos* φ and Eq. (3.16) by *-sin* φ and adding produces the result

$$\dot{\theta} = q\cos\varphi - r\sin\varphi \tag{3.17}$$

Substituting this result into Eq. (3.16) yields:

$$\dot{\psi} = q\frac{\sin\varphi}{\cos\varphi} + r\frac{\cos\varphi}{\cos\theta} = q\sec\theta\sin\varphi + r\sec\theta\cos\varphi \tag{3.18}$$

Finally, using this result in Eq. (3.14),

$$\dot{\varphi} = p + \dot{\psi}\sin\theta = p + q\tan\theta\sin\varphi + r\tan\theta\cos\varphi \tag{3.19}$$

In matrix form, these results can be rewritten as:

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \tan\theta\sin\varphi & \tan\theta\cos\varphi \\ 0 & \cos\varphi & -\sin\varphi \\ 0 & \sec\theta\sin\varphi & \sec\theta\cos\varphi \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} = T\begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{3.20}$$

where *sec* θ = 1 / *cos* θ. Evidently, this matrix is singular for $\theta = \pm 90°$. [Ref. 16.]

c. *Euler Angle Singularities*

When the nose unit vector of a rigid body points straight up (or down), the roll and azimuth gimbal axes are collinear. This means that neither roll or azimuth angles are uniquely defined, but rather, only their difference (nose up) or sum (nose down) can be specified uniquely [Ref. 51.]. This problem is also manifested in an even more serious way with respect to Euler angle rates since, the body rate to Euler rate transformation matrix (*T* in Eq. (3.20)) is singular for this orientation ($\theta = \pm 90°$). Obviously, this problem only arises for rigid bodies which are capable of assuming a vertical orientation.

2. Quaternions

Quaternions are an extension of complex numbers designed to define a four-dimensional volume using three “imaginary” parts and one “real” part. The imaginary portion of a quaternion is often termed the “vector” part. The real part of a quaternion is sometimes called the “scalar” part. Quaternions are commonly represented using three different notations.

(1) Linear combination of four components:

$$q = w + xi + yj + zk \tag{3.21}$$

where i, j, and k denote the standard orthonormal basis for three space.

(2) Four dimensional vector:

$$q = (w\ x\ y\ z) \tag{3.22}$$

(3) Scalar with a vector imaginary part:

$$q = (w, v) \tag{3.23}$$

It is also possible to write a quaternion as the sum of two four dimensional vectors. For a quaternion q, the vector $Re(q)$ contains the scalar or real part of q. Only the first element is nonzero. The vector $Ve(q)$ contains the vector or imaginary part of the quaternion. The first element is zero and the last three elements express a vector in component form. Thus,

$$q = Re(q) + Ve(q) = \begin{bmatrix} w \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ x \\ y \\ z \end{bmatrix} \tag{3.24}$$

Intuitively, the three imaginary parts describe a vector and the real part expresses an angle of rotation about that vector. The imaginary parts have the following properties

$$i * i = i^2 = -1 \tag{3.25}$$

$$j * j = j^2 = -1 \tag{3.26}$$

$$k * k = k^2 = -1 \tag{3.27}$$

and

$$ij = k = -ji \tag{3.28}$$

$$jk = i = -kj \tag{3.29}$$

$$ki = j = -ik \tag{3.30}$$

a. Quaternion Operations

Under the operations of addition and multiplication, quaternions satisfy all of the axioms of a field except the commutative law. Let s be a scalar and let

$$q_1 = w_1 + x_1 i + y_1 j + z_1 k \qquad q_2 = w_2 + x_2 i + y_2 j + z_2 k \tag{3.31}$$

The following operations are defined for quaternions.

Equality

Two quaternions are equal if and only if they have exactly the same components. That is q_1 and q_2 are equal if and only if

$$w_1 = w_2 \tag{3.32}$$

$$x_1 = x_2 \tag{3.33}$$

$$y_1 = y_2 \tag{3.34}$$

$$z_1 = z_2 \tag{3.35}$$

Addition

The sum of two quaternions is defined in the same manner as normal vector addition by adding corresponding components.

$$q_1 + q_2 = ((w_1 + w_2)(x_1 + x_2)(y_1 + y_2)(z_1 + z_2)) \tag{3.36}$$

Each quaternion q has a negative or additive inverse denoted by $-q$, in which each component is the negative of the corresponding component of q.

Scalar Multiplication

Scalar multiplication of a quaternion is commutative and is again defined in the same manner as that of a vector in four space. Each component of the quaternion is simply multiplied by the scalar.

$$sq = (sw, sv) \tag{3.37}$$

Quaternion Multiplication

The product associated with the multiplication of two quaternions is itself a quaternion.

$$q_1q_2 = (w_1 + x_1i + y_1j + z_1k)(w_2 + x_2i + y_2j + z_2k) \tag{3.38}$$

Using (3.25) through (3.30) and the distributive and commutative properties of scalar multiplication, (3.38) becomes

$$\begin{aligned} q_1q_2 = {} & (w_1w_2 - x_1x_2 - y_1y_2 - z_1z_2) \\ & + i(x_1w_2 + w_1x_2 - z_1y_2 + y_1z_2) \\ & + j(y_1w_2 + z_1x_2 + w_1y_2 - x_1z_2) \\ & + k(z_1w_2 - y_1x_2 + x_1y_2 + w_1z_2) \end{aligned} \tag{3.39}$$

The result given in (3.39) can also accomplished by scalar multiplication of the imaginary vectors, taking the dot products of the imaginary vectors (produces a scalar) and taking the cross product of the imaginary vectors (produces a vector). That is,

$$q_1q_2 = (w_1v_1)(w_2v_2) = (w_1w_2 - v_1 \cdot v_2, w_1v_2 + w_2v_1 + v_1 \times v_2) \tag{3.40}$$

Evaluation of Eq. (3.39) or Eq. (3.40) requires a total of 28 scalar operations (16 multiplies and 12 additions). Though quaternion multiplication is associative, the cross-product makes the operation non-commutative.

b. Quaternion Forms

Quaternion Conjugate

Let $q = (w, v)$, then the quaternion conjugate of q is

$$\bar{q} = (w, -v) \tag{3.41}$$

It can be shown that the conjugate of the product of the two quaternions is equal to the product of the individual quaternion conjugates in reverse order. That is

$$\overline{(q_1 q_2)} = \bar{q}_2 \bar{q}_1 \tag{3.42}$$

The sum of any quaternion and its conjugate will be the scalar quantity $2w$.

Norm

The norm of a quaternion is sometimes called the *length* or *magnitude* of the quaternion. Let $q = (w\ x\ y\ z)$, then the norm of q denoted $N(q)$ is

$$N(q) = \sqrt{\bar{q}q} = \sqrt{w^2 + x^2 + y^2 + z^2} \tag{3.43}$$

This definition is the same as the for the length of a four dimensional vector.

Normalized unit quaternion

If a quaternion has a norm of unity, each of its components must have an absolute value less than or equal to one. Such quaternions are called *unit* or *normalized* quaternions.

$$q_{normalized} = \frac{q}{N(q)} \tag{3.44}$$

Quaternion (multiplicative) inverse

In general, the multiplicative inverse of a quaternion q is given by

$$q^{-1} = \frac{q}{N^2(q)} = \frac{q}{\bar{q}q} \tag{3.45}$$

Since $N(q) = 1$ for a unit quaternion, the inverse of a unit quaternion is simply $\bar{q}$.

c. *Quaternion Transformation Between Coordinate Frames*

It is known that the orientation of a rigid body can always be described as a rotation (ϕ) about a single inclined axis (v). If the axis (v) is constrained to unit magnitude, the quaternion (q_r) representing this orientation is

$$q_r = \left(\cos\frac{\phi}{2}, v\sin\frac{\phi}{2}\right) \tag{3.46}$$

Evidently,

$$q_r\bar{q}_r = |q_r|^2 = 1 \tag{3.47}$$

The product of two unit quaternions is also of unit magnitude. The product q_1q_2 is a quaternion rotation from the orientation described by q_1 to a cumulative orientation of q_1 and q_2. In general, "earth coordinate" rotations multiply from the left and "body coordinate" rotations from the right. [Ref. 53.]

Any scalar can be represented as a quaternion.

$$q = (w\,0\,0\,0) \tag{3.48}$$

Any three dimensional point vector $p = (x\ y\ z)^T$can be represented as the quaternion with the real part set to zero.

$$p = (0\ x\ y\ z) \tag{3.49}$$

The rotation of a vector, p, by a quaternion q is defined as

$$p_{rotated} = qpq^{-1} \tag{3.50}$$

If q is of unit magnitude such that

$$q = \left(\cos\frac{\theta}{2}, u\sin\frac{\theta}{2}\right) \tag{3.51}$$

then

$$p_{rotated} = qp\bar{q} \tag{3.52}$$

where u is a unit vector about which the vector p is rotated through an angle θ. There are no singularities and only two trigometric functions involved. 56 scalar operations or twice the number needed to evaluate Eq. (3.40) are required to evaluate Eq. (3.52).

d. Unit Quaternions In Positive Real Form

For any given orientation, there are two unit quaternions which may be used to represent it. The unit quaternions

$$q_1 = \begin{bmatrix} w \\ x \\ y \\ z \end{bmatrix} \qquad q_2 = -q_1 = \begin{bmatrix} -w \\ -x \\ -y \\ -z \end{bmatrix} \tag{3.53}$$

both represent the same orientation. To eliminate this ambiguity and insure a unique solution for an orientation, the angle of rotation, α, may be restricted to $-\pi/2 \leq \alpha \leq \pi/2$. Since the real part of such a quaternion will always be positive, it can be recovered using the assumption of unit length. Thus, the four elements of a unit quaternion in this *positive real form* are not independent. For such a quaternion $q = [\ w\ x\ y\ z\ ]$

$$w = \sqrt{1 - x^2 - y^2 - z^2} \tag{3.54}$$

from which it follows that

$$x^2 + y^2 + z^2 = 1 - w^2 \tag{3.55}$$

and

$$x^2 + y^2 + z^2 + w^2 = 1 \tag{3.56}$$

If w is allowed to vary between negative and positive one in Eq. (3.55) and Eq. (3.56), these equations become descriptions of the interior and surface of a unit sphere in three-dimensional space. This sphere is filled twice, once as w varies between 0 and 1, and once as w varies between 0 and -1.

e. Transforming Angular Rates To A Quaternion Rate

Angular rates, p, q, and r, may be used to find the derivative of the orientation quaternion, $\dot{q}$, relative to the earth-fixed coordinate system. Suppose a rigid body is first rotated by an angle θ_1 about an inclined axis specified by the unit vector v_1. If v_1 is in earth coordinates, the unit quaternion representing this rotation is

$$q_1 = \left(\cos\frac{\theta_1}{2},\ v_1 \sin\frac{\theta_1}{2}\right) \tag{3.57}$$

Assume the body is then rotated by an angle θ_2 about a second axis v_2 expressed in body coordinates by the unit quaternion q_2.

$$q_2 = \left(\cos\frac{\theta_2}{2},\ v_2 \sin\frac{\theta_2}{2}\right) \tag{3.58}$$

For small θ_2,

$$\cos\frac{\theta_2}{2} \cong 1 \qquad \sin\frac{\theta_2}{2} \cong \frac{\theta_2}{2} \tag{3.59}$$

and thus (3.58) becomes

$$q_2 \cong \left(1,\ v_2\frac{\theta_2}{2}\right) \tag{3.60}$$

Assuming θ_2 changes linearly with time, the orientation expressed by q_2 as a function of time becomes

$$q_2(t) = \left(1,\ \frac{1}{2}v_2\dot{\theta}_2 t\right) \tag{3.61}$$

for small t. $v_2\dot{\theta}$ expresses an angular rate of $\dot{\theta}$ about a vector v_2 in body coordinates. Thus

$$v_2\dot{\theta} = (0\ \ p\ \ q\ \ r) \tag{3.62}$$

and (3.61) becomes

$$q_2(t) = \left(1,\ \frac{1}{2}pt\ \ \frac{1}{2}qt\ \ \frac{1}{2}rt\right) \tag{3.63}$$

Taking the derivative of (3.63) with respect to time produces

$$\frac{d}{dt}q_2(t) = \dot{q} = \left(0,\ \frac{1}{2}p\ \ \frac{1}{2}q\ \ \frac{1}{2}r\right) \tag{3.64}$$

$$= \frac{1}{2}(0,\ \ p\ \ q\ \ r) \tag{3.65}$$

$$= \frac{1}{2}{}^{B}\omega \tag{3.66}$$

If q_1 is the initial orientation is earth coordinates and q_2 is a second rotation in body coordinates, then q_3 is the composite rotation combining the two rotations.

$$q_3 = q_1 q_2 \tag{3.67}$$

By the product rule

$$\dot{q}_3 = \dot{q}_1 q_2 + q_1 \dot{q}_2 = q_1 \dot{q}_2 = \frac{1}{2} q_1 {}^B\omega \tag{3.68}$$

The components of $\dot{q}_3$ are given by

$$\dot{q}_{30} = -\frac{1}{2}(q_{11}p + q_{12}q + q_{13}r) \tag{3.69}$$

$$\dot{q}_{31} = \frac{1}{2}(q_{10}p + q_{12}r - q_{13}q) \tag{3.70}$$

$$\dot{q}_{32} = \frac{1}{2}(q_{10}q + q_{13}p - q_{11}r) \tag{3.71}$$

$$\dot{q}_{33} = \frac{1}{2}(q_{10}r + q_{11}q - q_{12}p) \tag{3.72}$$

In general, Eq. (3.69) through (3.72) are expressed by the quaternion multiplication

$$\dot{q} = \frac{1}{2} q {}^B\omega = \frac{1}{2} q(0 \;\; p \;\; q \;\; r) \tag{3.73}$$

Note that Eq. (3.73) offers the potential of orientation tracking of rigid bodies using no trigometric functions whatsoever. [Ref. 53.]

f. Representing Orientations Without Singularities

Quaternions can be used to represent all orientations without singularities and thus are a logical choice when representing the orientation or a rigid body which may go through the vertical. A precise method of overcoming the singularities associated with Eq. (3.20) involves transformation of rotational rates sensed in body coordinates into a rate quaternion Eq. (3.73), and integrating to get a quaternion representation of orientation [Ref. 52.].

C. MODELS FOR HUMAN BODY TRACKING

Unlike dynamics models, kinematic models involve the study of motion independent of the underlying forces which cause it. Only geometrical and time related properties of motion such as position, velocity and acceleration are defined. [Ref. 17.] Kinematic models represent articulated structures as a series of interconnected links. The relationships between these links may be extremely complex. They may be described using either homogenous transformation matrices or quaternion/vector pairs.[Ref. 30.]

1. Kinematic Models Based On Homogenous Transformation Matrices

The human body can be modeled as an articulated structure involving links connected by revolute joints. Multiple degree of freedom joints can be modeled as multiple collocated single degree of freedom joints. Each joint is assigned an individual reference frame which is related to the reference frames of adjacent links by a 4 x 4 homogenous transformation matrix which expresses both a rotation and a translation. The rules used to derive the matrix are dependent on the notation in use. There are two common, but similar notations for expressing the relationship between neighboring joints in the an articulated structure. These notations are the *Denavit-Hartenberg* (DH) and the *Craig* or *Modified Denavit-Hartenberg* (MDH). The DH and MDH notations are equivalent, with the exception that the link frame of reference coordinate origin for DH links is attached to the outboard motion axis of the link while the corresponding origin for MDH links is attached to the inboard motion axis. [Ref. 17.]

As a body moves, the relationships between the frames associated with the links change. Thus, describing a body posture simply involves expressing the relationships between adjacent frames. Four parameters are used to describe the relationship. These are *link length*, *link twist*, *link offset*, and *joint angle*. In an articulated structure involving only revolute joints, only changes in joint angle occur. All other parameters are fixed.

Figure 3 depicts frame assignment and the standard MDH parameters associated with each link. Link$_{i-1}$ is inboard of axis$_i$ and thus Link$_{\mathrm{i-1}}$ is referred to as the inboard link and link$_i$ as the outboard link. Again DH is equivalent, but attaches the link frame of reference to the outboard motion axis. The four MDH parameters depicted are:

- inboard link length: a_{i-1} = distance from z_{i-1} to z_i measured along x_{i-1}
- inboard link twist: α_{i-1} = angle between z_{i-1} and z_i measured about x_{i-1}
- outboard link offset: d_i = distance from x_{i-1} to x_i measured along z_i
- outboard joint angle: Θ_i = angle between x_{i-1} to x_i measured about z_i

Once the link parameters have been measured, a MDH transformation matrix which relates the frame for i-1 to that of i can be created. It is given below by [Ref. 17.]:

$$ {}^{i-1}T_i = \begin{bmatrix} \cos\Theta_i & -\sin\Theta_i & 0 & a_{i-1} \\ \sin\Theta_i\cos(\alpha_{i-1}) & \cos\Theta_i\cos(\alpha_{i-1}) & -\sin(\alpha_{i-1}) & -\sin(\alpha_{i-1})d_i \\ \sin\Theta_i\sin(\alpha_{i-1}) & \cos\Theta_i\sin(\alpha_{i-1}) & \cos(\alpha_{i-1}) & \cos(\alpha_{i-1})d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.74} $$

Rotating and positioning the outboard joint coordinate system relative to the inboard joint coordinate system requires multiplication of ${}^{i-1}T_i$ and ${}^{i}T_{i+1}$. This composition of two 4 x 4 matrices will require 64 multiplications and 48 additions Taking into account the redundant last rows this can be reduced to 36 multiplies and 27 additions [Ref. 88.].

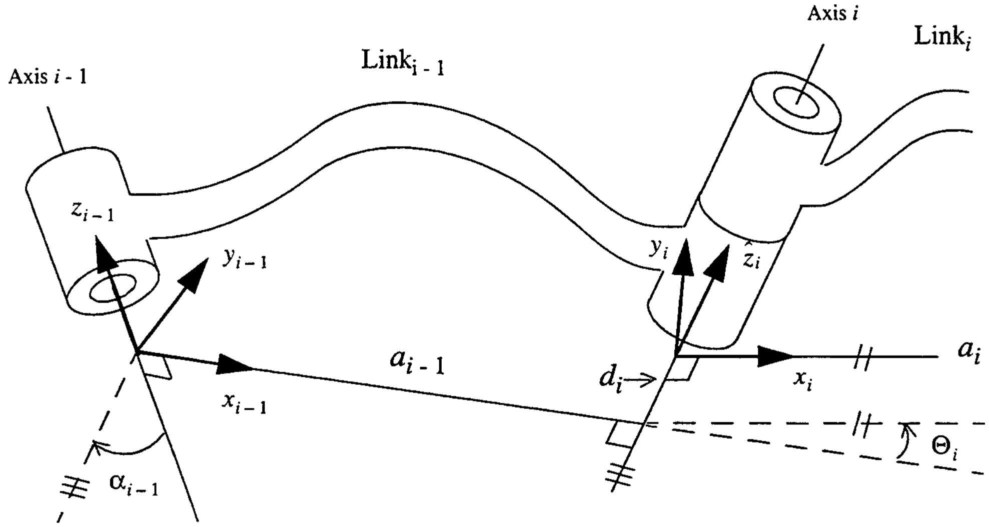


Figure 3: Frame Assignment Under MDH After [Ref. 17.]

2. Forward and Inverse Kinematics

Kinematic problems are often separated into two classes. In *forward kinematics* the motion of the end effector is determined indirectly as the accumulation of the transformations that lead to it. All joint angles are specified explicitly to define an exact position for the entire structure. Complete control is maintained over the kinematic structure, but it may be counterintuitive and complicated to use in practice. Forward kinematics applications are less demanding computationaly and are commonly used to set predefined postures. *Inverse kinematics* or *goal directed* motion entails calculating joint angles given the position and orientation of the last link or end-effector and possibly some intermediate links. Inverse kinematics is generally not as straightforward as forward kinematics. In an *under specified* system involving a minimal amount of information, there may be an infinite number of solutions for a given end-effector position and orientation. As the number of links increases, the difficulty of finding an unambiguous solution will increase. In such a case, additional constraints or heuristics may be applied to the system to allow a unique solution to be selected. The solution may be closed form or it may be arrived at numerically. Methods of obtaining closed form solutions may be either algebraic or geometric. In any event, the method of solution will tend to be unique for each specific case. Performing the computations involved in inverse kinematics in real-time can often be difficult even when using a closed form solution.

Minimizing the number of position/orientation sensors used in body tracking applications reduces user encumbrance. However, reducing the number of sensors can mean that the orientation of some links will not be tracked directly. Since solving the inverse kinematics problem allows the transforms for untracked links which are not directly sensed to be found, a great deal of research has been done involving inverse kinematics in body tracking applications [Ref. 10.][Ref. 11.][Ref. 85.][Ref. 79.]. In these applications, 6 DOF sensors are required on all end-effectors such as the hands and feet. The solutions found require considerable computational overhead. In addition, the posture of the model can be inaccurate if the heuristics and constraints employed cause the selection of a solution that does not match the actual position of the user.

3. Kinematic Models of the Human Body based on Joint Angles

Kinematic models of the human body are often quite complex. Attempts to simulate the range of motion of the human skeleton typically result in articulated structures containing on the order of 60 degrees of freedom [Ref. 11.][Ref. 78.][Ref. 88.]. Such models can require a transformation matrix as given by Eq. (3.74) for each link. Since links can not be positioned independently with such a model, each change in posture requires up to 60 matrix multiplications or at least 3,780 scalar operations to reposition the model. The computational load will increase further if there is a need for inverse kinematic calculations to determine the positions of some limbs. Nevertheless, kinematic models based on homogenous transformation matrices and joint angles are well suited for use with sensing systems which provide joint angles as output. Other alternative models may be more appropriate when working with sensors which provide link orientation and/or position relative to Earth coordinates as output.

Noisy or inaccurate sensor information in human body tracking applications can result in postures which are unrealistic of impossible for a human to perform. For example, when the elbow is completely flexed, inaccuracies in sensor data due to noise or a lack of precision can place the upper and fore arms in the same location. Joint angle models based on transform matrices allow the implementation of joint limits which match the motion limits of a human skeleton. If sensor data results in a calculated position which is beyond the joint limits of the link, the limb can simply be placed at the limit and transformations can continue based on this "limited" position. Often other representations of limb segment orientation are converted to matrix form for this purpose. In [Ref. 88.], orientation data is input in quaternion form. These quaternions are then turned into rotation matrices for application of joint constraints and submission to the graphics API.

In networked applications involving body tracking, it is necessary to pass posture data between remote locations. If full kinematic models containing all fixed transformation matrix parameters are stored at each location, only the variable joint angles need to be sent across the network each time a posture update is made. Thus, if a sixty DOF model is used and the joint angles are specified using 16 bit numbers, only 120 bytes of information must be sent across the network. It should be noted, however, that once the joint angle data has been received, each location will be required to perform numerous matrix multiplications to reposition the model. It would thus be desirable to find a method of specifying orientation with an equivalent network bandwidth requirement that required a more limited computational overhead at each location.

4. Orientation Only Tracking

In orientation only tracking applications, the posture of a human model is set using only orientation data. Position data for a single reference point is used only to place the entire human model within a synthetic environment.

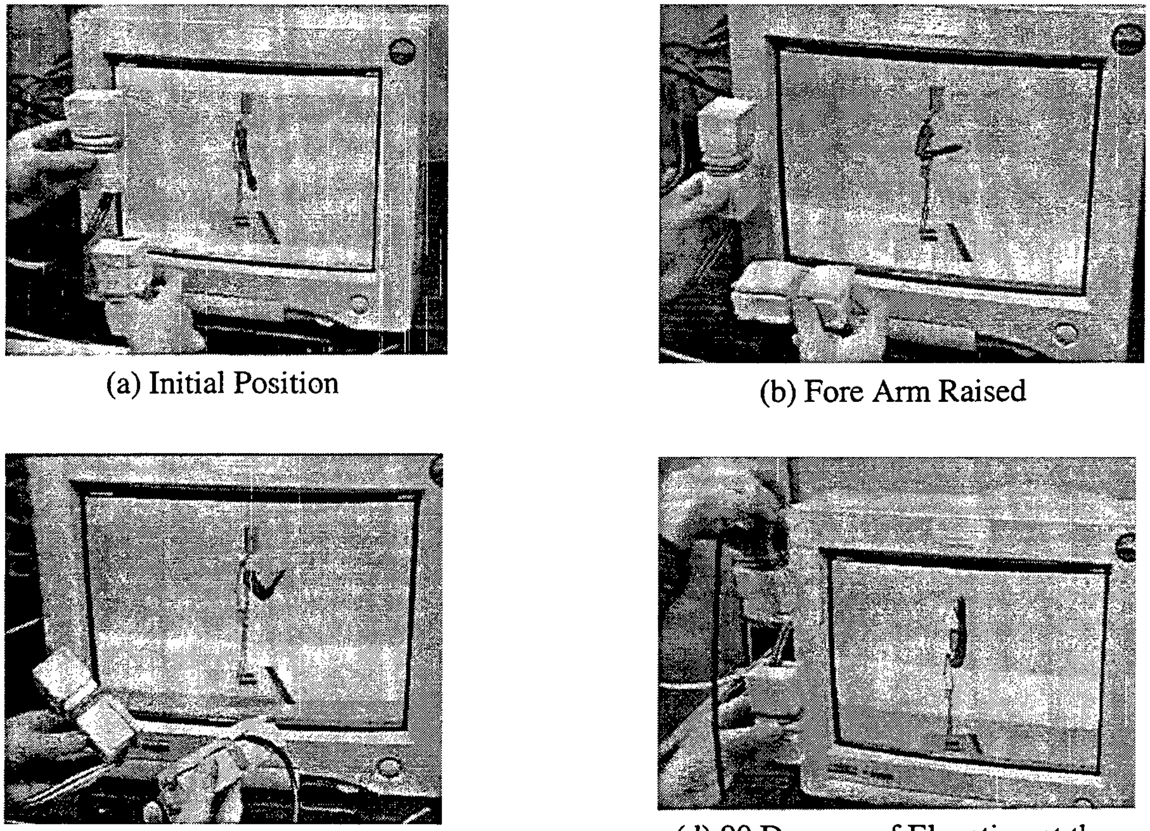
Inertial sensors provide orientation relative to an earth fixed coordinate reference frame. In early inertial angle tracking work in [Ref. 28.], Frey showed that an entire human body simulation can be built and animated using only orientation data for each body part. This result eliminated the need for human body motion capture systems to track the position of each body part and showed that orientation data alone could be used to determine body posture.

Usta created a human model designed to accept a quaternion representation of orientation relative to an earth fixed coordinate reference frame. The input data was provided by prototype inertial trackers. The quaternions were then turned into rotation matrices for submission to the graphics API and the application of joint constraints [Ref. 88.]. He did not use the quaternions to directly orient individual body segments for graphical rendering. Qualitative results from his work are shown in Figure 4. Only static tests were performed.

Other work has discarded the position data from active magnetic systems for posture determination and used only orientation data to drive the animation of a human model. This orientation data was used to determine joint angles which were applied to kinematic models [Ref. 78.], [Ref. 64.]. Though Molet transmitted orientation quaternions across a network to save bandwidth, the quaternions were converted to rotation matrices. Inverse kinematic calculations were made to allow several joints to be driven with one sensor [Ref. 64.].

5. Kinematic Models based on Quaternion/Vector Pairs

Quaternion/vector pairs represent a rotation using a quaternion and a translation using a vector. [Ref. 30.] Utilization of sensors which output orientation data in an earth-



(a) Initial Position

(b) Fore Arm Raised

(c) Forearm and Upper Arm

(d) 90 Degrees of Elevation at the shoulder (No singularity)

Figure 4: Inertial Motion Tracking of the Right Fore and Upper Arm with Two Inertial Sensors and a Quaternion Attitude Filter From [Ref. 88.]

fixed coordinate reference frame is more applicable to the use of kinematics models based upon quaternion/vector pairs. In this case, each limb segment can be oriented without regard to the orientation of adjacent segments [Ref. 28.]. The posture of the user can be reconstructed by simply attaching the representations of individual limb segments together in the same manner in which the corresponding segments on the body of the user are connected. There is no need for coordinate transformations or the associated transformation matrices to determine joint angles. Body posture is entirely determined based upon limb orientation and length and the quaternion and vector which represent these parameters.

Given low noise orientation data of sufficient accuracy, it should not be necessary to apply joint angle constraints to correct position errors. If this data is supplied in quaternion form, the need to generate rotation matrices and perform numerous matrix multiplications can be avoided. Each limb segment can be oriented via multiplication by unit quaternions as described by Eq. (3.52). Limb segments can be positioned through a rotated translation vector derived from concatenation of vectors pointed from proximal to distal joints.

Figure 5 depicts a human model designed for the input of quaternions representing the orientations of the individual limb segments. The animation of the human figure is accomplished without rotation matrices. When all of the limb segments are in their reference positions, the body-referenced x axes are pointing north, y axes pointing east and z axes point down. The orientation of each limb segment in its reference position is described by the unit quaternion

$$q = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{3.75}$$

The first element of this quaternion is the cosine of the half angle of rotation. When in the reference position there is no rotation (*cos* 0 = 1).

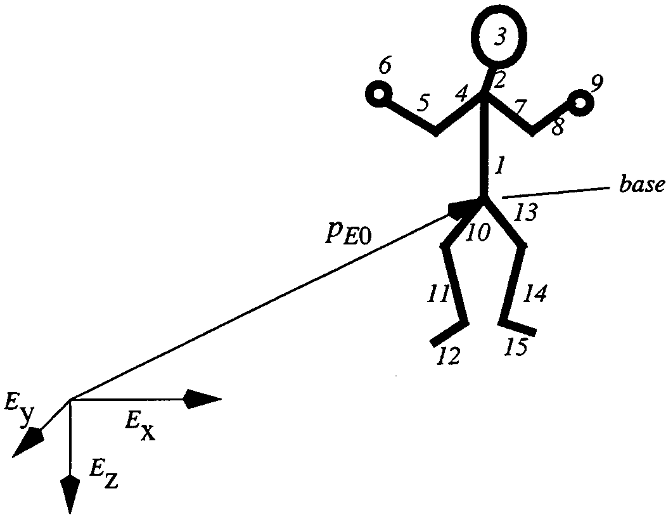


Figure 5: Human Model Designed For Quaternion Input

Each numbered link in Figure 5 has two connection points and a vector which connects them. This vector direction is outboard toward the more distal joints. The vector $\boldsymbol{p}_{ij}$ extends from the inboard connection point to the outboard connection point of link j. The vector $\boldsymbol{p}_{ij}$ is a quaternion with a real part equal to zero. The length of $\boldsymbol{p}_{ij}$ is equal to the length of link j. Thus, the position and orientation of limb segment j is described by the quaternion/ vector pair $(q_j, \boldsymbol{p}_{ij})$.

When a link j is no longer in its reference position, the orientation of that link is given by the unit quaternion q_j. Thus, the orientation transformation applied to each vertex, v_j, in the graphical representation of the limb segment corresponding to link j is

$$v'_j = q_j v_j \bar{q}_j \tag{3.76}$$

For link 1 the rendered position and orientation is given by vertex transformation

$$v'_1 = p_{E0} + q_1 v_1 \bar{q}_1 \tag{3.77}$$

For link 2 the transformation applied to each vertex is

$$v'_2 = p_{E0} + q_1 p_{01} \bar{q}_1 + q_2 v_2 \bar{q}_2 \tag{3.78}$$

In general, the nth link outboard from the base is positioned and oriented by

$$v'_n = p_{E0} + q_1 p_{01} \bar{q}_1 + q_2 p_{12} \bar{q}_2 + \ldots + q_{n-1} p_{n-1n} \bar{q}_{n-1} + q_n v_n \bar{q}_n \tag{3.79}$$

Obviously, the links should be positioned and oriented by working outward from the base and saving intermediate results. This eliminates the need to repeat identical calculations when multiple limb segments are attached to the same inboard link. It should be noted that Eq. (3.76) through Eq. (3.79) involve only scalar additions and multiplications. The are no trigometric functions or matrix multiplications. Positioning and orienting the structure depicted in Figure 5 will require approximately 840 scalar operations.

In networked simulations, the use of quaternions requires considerably less bandwidth than that of joint angles. Specifically, for a unit quaternion all elements are within the range +/- 1. Integer representation of a unit quaternion with 1% accuracy therefore requires four bytes. For the purpose of networked simulation, the human body can be adequately modeled using fifteen limb segments. Thus, posture updates using a quaternion representation require that approximately 60 bytes of information be sent across the network. This is roughly the equivalent of that required for the update of a kinematic model with 60 degrees of freedom. If unit quaternions in positive real form are used, only 45 bytes need be transmitted. In addition, the amount of computation at each end will be greatly reduced when quaternion representations of orientation are used.

D. SUMMARY AND CONCLUSIONS

Homogenous transform matrices and Euler angles are widely used by both the graphics and robotics communities. This utilization is mostly due to their familiarity and matrix formulation. There are however several disadvantages to the use of these formalisms. Homogenous transform matrices require the storage of 16 numbers, seven of which are redundant or constant for any matrix. The composition of two rotation matrices requires 36 scalar multiplications and 27 adds. The use of trigometric functions is even more expensive since approximation is usually carried out using Taylor series. Within a rotation matrix there are at least six trigometric functions which must be evaluated. Each requires numerous scalar operations. Use of Euler angles results in singularities whenever the inner and outer gimbal rotation axes become collinear. Thus, they are not appropriate for tracking the orientation of a rigid body that can assume any orientation.

Kinematic models of articulated structures which are based on homogeneous transform matrices must orient and position each link with respect to the orientation and position of the inboard connecting link. Changing the posture of a 60 DOF human model will require at least one multiplication of two 4 x 4 matrices per joint. Kinematic models based on homogeneous transform matrices are well suited to tracking systems which provide joint angle output. The ability to implement joint limits allows correction of some problems which might occur when using noisy or inaccurate sensors.

Quaternions and quaternion/vector pairs offer an alternative to rotation matrices based on Euler angles and homogenous transform matrices.[Ref. 30.] Though in less general use, in terms of computational efficiency and compactness they are superior. Thinking about a matrix which expresses a rotation about a non-principal axis is just as difficult as imagining a quaternion which specifies a rotation about an arbitrary vector. Thus, intuitively quaternions are no more difficult to work with than rotation matrices. Storage of a quaternion requires four numbers (three for quaternions in positive real form) where as an equivalent rotation matrix requires nine. Quaternion vector pairs require the storage of only seven numbers in contrast to the sixteen of a homogenous transform matrix. The composition of two rotations and translations using quaternion/vector pairs requires only 32 scalar multiplications and 24 additions. In many practical applications, there is no need to evaluate any trigometric functions. Quaternion representations of orientation do not result in any singularities.

Kinematic models based on quaternion/vector pairs are computationaly more efficient than those based on homogeneous transform matrices. This is especially true when they are driven by orientation data which is described relative to a world coordinate reference frame. When compared with joint angle updates of posture, the bandwidth requirements are roughly the same. If it is necessary to transmit both translation and orientation data, quaternion/vector pairs require approximately one fifth the bandwidth of homogenous transform matrices. Update of the posture of a 15 segment human model will require 840 scalar operations. This is an order of magnitude less than the 3,780 scalar operation needed to reset the posture using transform matrices. An articulated structure based on quaternion vector pairs includes no notion of joint angles. Thus, it is not possible to implement joint angle constraints using this formalism and when using noisy or inaccurate sensors it may be advisable to adopt the more traditional approach of a Denavit-Hartenberg type system.

IV. REVIEW OF FILTER THEORY AND DESIGN

A. INTRODUCTION

In physical applications, sensor outputs are commonly processed by digital computers with the intention of making some determination regarding the physical world. Examples of these determinations may include estimates of velocity, acceleration, position, temperature, pressure, etc. In human body tracking applications, the goal is to use sensor signals to estimate the orientation of a rigid-body. Unfortunately, because of size limitations and cost considerations, sensor output is rarely of sufficient quality to allow direct estimation using naive algorithms [Ref. 49.]. The sensors themselves will have accuracy limitations. In addition, the output of the sensors will be corrupted by noise. Thus, it is necessary to process sensor output data in a more rigorous manner to separate the actual sensor signal from the noise which is present and arrive at the "best" estimate possible given the inaccuracy of the sensors themselves. The algorithms used to process the signals from the sensors are generally termed *filtering algorithms.*

The primary purpose of a filter or filtering algorithm is to separate signals from noise. Classic examples of this type of filtering include high and low pass filters which respectively attempt to separate low and high frequency noise from a signal. Removal of noise from a signal will tend to smooth the output. More sophisticated filtering may also combine signals from several sensors in order to produce an estimate which is "optimal" with respect to some criteria. These types of filters are usually based upon a probabilistic model of the signal being estimated as well as the overall system to which it is related. Encapsulation of this model within the algorithm provides the additional capability of prediction. This may be important in applications in which timeliness is critical, since a predicted value can be used in place of an actual estimate.

Inertial/magnetic human body tracking is essentially a navigation problem with the goal of determining the orientation of each body segment. Sensor input comes from miniaturized sensors. No single input is of sufficient quality to accurately determine orientation over a long period. It is therefore necessary to combine the signal from several sensors to arrive at an accurate estimate of orientation. Since the effects of lag are so devastating to the sense of presence in a synthetic environment, calculation of this estimate must not be so computationally demanding that it can not be made in real time. Thus, the most accurate filter possible may not be the best choice if it is too slow.

B. MINIATURE INERTIAL SENSORS

MEMS, or microelectromechanical systems, are integrated systems combining both electrical and mechanical components. Unlike conventional semiconductor manufacturing or microelectronics in which electronic circuits are implemented, MEMs devices contain three dimensional mechanical structures. These "micromachined" mechanical structures have dimensions which are measured in micrometers. By combining microelectronics and micromachining, precision electronics are closely integrated on the same device. The electronics sense the positions and deflections of the mechanical elements. Since they are in such close proximity, parasitics and noise are reduced and reliability is improved.
[Ref. 9.]

At least four different micromachining techniques are in use or under development. *Silicon micromachining* is a relatively developed micromachining technique since it is closely related to the production of microelectronic circuitry. Silicon is the primary substrate material used. *Electrochemical etching techniques* are being investigated to extend the set of basic silicon micromachining techniques. *Silicon bonding techniques* can also be utilized to extend the structures produced by silicon micromachining techniques into multiple layers. *Excimer laser techniques* use an ultraviolet laser to micromachine a number of materials without heating them. The excimer laser lends itself particularly to the machining of organic materials (plastics, polymers, etc.). LIGA[1] is a technique that can be used to produce molds for the fabrication of micromachined components. Microengineered

[1] The acronym LIGA comes from the German name for the process (Lithographie, Galvanoformung, Abformung).

components can be made from a variety of materials using this technique. More than one micromachining technique can be involved in the manufacture of a hybrid MEMs device. Photolithography is used in conjunction with all of the micromachining techniques described above. [Ref. 9.]

Sensors are a specialized type of transducer. MEMs sensors convert a physical or chemical quantity into an electrical one. Though each sensor type has a set of advantages and disadvantages, the orientation of a rigid body may be determined using only data from body-mounted accelerometers, angular rate sensors, or magnetometers. Improved static and dynamic accuracy can be obtained by combining data from all of the sensor types in a complementary manner.

MEMs magnetic sensors or magnetometers can use several different methods to sense the local magnetic field. Hall effect sensors consist of a conducting material, usually a semiconductor, through which a current is passed. In these sensors, changes in anisotropic magnetoresistance (AMR) occur when a magnetic field is applied perpendicular to the current flow. Two magnetoresistive sensing elements or contacts may be placed on opposite corners of the device. Sensing contacts are also placed on the remaining corners of the device, opposite each other and perpendicular to the current flow. Changes in the magnetic field perpendicular to the plane of the contacts are detected as a change in the potential difference between the two sensing contacts. [Ref. 9.][Ref. 43.]

Several major techniques are used to design MEMs accelerometers. Due to the newness of the field, performance ranges and optimal application areas of each have yet to be determined. In one technique, a silicon diaphragm to which a mass has been added is the basic structure used. Under acceleration, the diaphragm bends causing a change in the distance between a stationary and moving electrode. The resulting change in capacitance is converted into a voltage. Piezo resistive materials in which the resistance changes as the material bends can also be used. Accelerometers based on this technique, consist of a mass suspended from thin beams. Under acceleration, a force ($f = ma$) is developed which bends the suspending beams. Piezoresistors positioned where the beams meet the support are used to detect acceleration. Vibrating beam accelerometers (VBAs) use two resonators vibrating at their natural frequency. One resonator is compressed by acceleration while the other is tensioned. The frequency of the tensioned resonator increases while the frequency of the compressed resonator decreases. The acceleration is determined by the difference between the two frequencies. [Ref. 9.][Ref. 43.]

Most miniature and micromachined angular rate sensors are based on the Coriolis tuning fork principle. Gyroscopes that use vibrating rather than rotating bodies to detect gyroscopic torques from coriolis acceleration are more reliable and less expensive than rotating gyros. The "tuning fork" structure is set to stable vibration at its fundamental frequency. As it is rotated about its axis, Coriolis acceleration generates a sinusoidally varying precession. The amplitude of the generated sine wave is proportional to the input angular rate about the axis and is given by

$$a = 4v\frac{\Omega}{K} \tag{4.1}$$

where v is the tine velocity, Ω is the input rate, and K is the stem torsional stiffness constant. [Ref. 43.]

C. RANDOM PROCESSES

In order to work with the output of a sensor, it is necessary that its output signal be described in mathematical terms. In filter theory, the characteristics of a signal are captured by the notion of the *stochastic* or *random process*. The concept of a random process associates time with a random variable. In this abstraction, it is imagined that an *ensemble* of identical experiments are conducted simultaneously. In each of these experiments the random signal of interest is being generated. The value or *state* of a random process, X, can be examined at any time t. For a fixed time t, the value of the random process is described by the random variable, x.

$$X(t) = x \tag{4.2}$$

If the parameter t is discrete, then X is a discrete-time random process. If the parameter t is continuous, then X is a continuous-time random process. Since X is random, the value at time t will generally not be the same for all experiments. What is of interest is the *expected value*, and how the process is correlated with itself in time, and how the process might be correlated with other processes in time.

How a process is correlated with itself in time is expressed by the *autocorrelation function*. The relationship between the state at times t_1 and t_2 is given by

$$R_X(t_1, t_2) = E[X(t_1)X(t_2)] = E[x_1x_2] = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x_1x_2f(x_1, x_2, t_1, t_2)dx_1dx_2 \tag{4.3}$$

where f is the second order probability density function for X. If a process is closely correlated with itself, the value of Eq. (4.3) will be positive. If Eq. (4.3) has a value of zero, the process is uncorellated with itself in time. For a *stationary process*, the value of Eq. (4.3) is only dependent upon the difference, $\tau = |t_1 - t_2|$. The *power* of a signal is given by the autocorrelation function when $t_1 = t_2$. That is,

$$R_X(t, t) = E[X(t)X(t)] = E[X^2(t)] \tag{4.4}$$

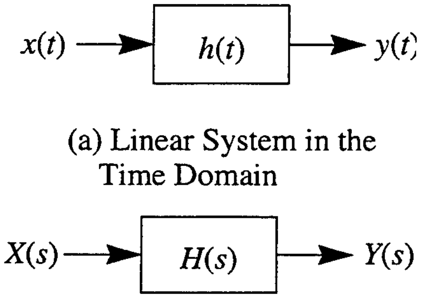
The relationship between two process is expressed by the *crosscorrelation function*. The correlation between the process X and Y at the times t_1 and t_2 is given by

$$R_{XY}(t_1, t_2) = E[X(t_1)Y(t_2)] = E[x_1y_2] = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x_1y_2f(x_1, y_2, t_1, t_2)dx_1dy_2 \tag{4.5}$$

Again, if the processes are uncorellated Eq. (4.5) will have a value of zero. Negative values indicate the processes are negatively correlated. *Autocovariance* and *crosscovariance* are zero mean versions of the autocorrelation and crosscorrelation functions respectively.

In filter theory, both the input and the output of a filter or system are treated as random processes [Ref. 14.]. Thus, filter design becomes an in depth examination of how the input processes of a system are related to the output processes. In practical applications, several simplifying assumptions are usually made. Processes are usually treated as *Wide Sense Stationary* (WWS) meaning that second order probability density functions are *time invariant*. It is also usually assumed that processes are *ergodic* or only a single sample time signal of a process is needed to obtain all necessary information about the signal. Proving two processes are *independent* requires that any moment of their joint probability density functions will be zero. This is usually very difficult. Thus, independence is usually only assumed based upon empirical data [Ref. 14.].

D. LEAST SQUARES FILTERING



(a) Linear System in the Time Domain

(b) Linear System in the Frequency Domain

Figure 6: Block Diagrams of Linear Systems

A system is a mathematical model that relates an input signal *x* to the output signal *y*. Figure 6 shows block diagrams of *linear systems* in the time and frequency domains. In each system diagram, the input is related to the output by a function. When working in the time domain, this function is call the *impulse response* (*h*(*t*) in diagram (a)). The relating function in the frequency domain is termed the *transfer function* (*H*(*s*) in diagram (b)). Mathematically, the output or a linear system is expressed by a *convolution integral*. In the time domain the integral is written

$$y(t) = \int_{-\infty}^{\infty} h(\tau)x(t-\tau)d\tau = \int_{-\infty}^{\infty} h(t-\tau)x(\tau)d\tau \qquad (4.6)$$

In the frequency domain, the convolution integral becomes a simple multiplication.

$$Y(s) = H(s)X(s) \qquad (4.7)$$

It should be noted, that working in the frequency domain is specialized to *WSS* processes [Ref. 14.].

If the input and the outputs of a linear system are random processes, it becomes necessary to determine the expected output of the system given an expected input. In the time domain for a linear system Eq. (4.6) becomes

$$
\begin{aligned}
E[Y(t)] &= E\left[\int_{-\infty}^{\infty} h(\tau)X(t-\tau)d\tau\right] \\
&= \int_{-\infty}^{\infty} h(\tau)E[X(t-\tau)]d\tau
\end{aligned}
\tag{4.8}
$$

Filters are described as systems in which the inputs and outputs are random processes. In a linear system, the impulse response or transfer function is applied to a noisy input to produce an estimate of a desired noiseless output, which is written

$$
\hat{Y}(t) = \int_{-\infty}^{\infty} h(\tau)X(t-\tau)d\tau \tag{4.9}
$$

The purpose of the filter under least square filtering is to minimize the estimation error. In particular, if the estimate of the noiseless output, $Y(t)$, is perfect the difference between these two values will be zero. Minimization of the square of the expected error takes the form

$$
e = E\{[Y(t) - \hat{Y}(t)]^2\} \tag{4.10}
$$

where *e* is the *squared error criterion*. In *linear minimum mean-square error estimation*, it is assumed that *X*(*t*) and *Y*(*t*) are related to one another by some linear function. Eq. (4.9) replaces $\hat{Y}(t)$ in Eq. (4.10) with a term involving the filter impulse response and the input.

$$
e = E\left\{\left[Y(t) - \int_{-\infty}^{\infty} h(\tau)X(t-\tau)d\tau\right]^2\right\} \tag{4.11}
$$

In the method of *nonlinear mean-square estimation* it is assumed that the input and output processes are related by a nonlinear function. In this case, the squared error criterion becomes

$$
e = E\{[Y - g(X)]^2\} \tag{4.12}
$$

where g is some nonlinear function of X. Choosing the form of the function g is difficult due to the multitude of possibilities. In theory, the best nonlinear estimator is given by

$$g(X) = E[Y|X] \tag{4.13}$$

However, in most practical applications this function is difficult to find [Ref. 58.]. In many cases, a more tractable problem can be created by approximating a nonlinear relationship using a linear function.

The exact manner in which the transfer function is determined is what characterizes the different types of filters. Wiener filters are linear mean square error filters for stationary random processes. Complementary filters are a specialization of Wiener filters in which no assumptions are made about the signal structure. Kalman filters are also linear mean square error filters in which the estimation process is recursive. The process model of an *Extended Kalman filter* is nonlinear, but the estimation itself is linear. [Ref. 14.]

E. WIENER FILTERING

Linear mean square error filtering began with the work of Nobert Wiener. [Ref. 14.] This work attempted to separate one *noiselike* signal from another. The end result tells how past values of input should be weighted in order to estimate the present value of the output. The theory developed is characterized by the following assumptions [Ref. 14.]:

- Both the signal and noise are random processes with known auto- and crosscorrelation functions.
- The criterion for best performance is minimum mean-square error.
- A solution based upon scalar methods will lead to the optimal filter weighting function.

The significance of the first and third assumptions should be noted. The first indicates that the complete spectral characteristics of both the noise and the signal must be known. The exact manner in which all signals are related must also be known in order for a Weiner filter to produce an optimal estimate. The third assumption emphasizes the reliance of the Weiner filter theory upon scalar methods. This reliance makes it difficult to apply Wiener filter theory to systems with multiple time inputs and outputs. Wiener filters may be either continuous or discrete.

1. Continuous Weiner Filters

If it is assumed that all processes are stationary and the filter is not time-varying, prediction, filtering and smoothing problems may be solved with a Weiner filter. If the input signal is continuous, the Wiener filter estimate of the output at a particular time t is formulated as

$$\hat{Y}(t) = \int_a^b h(t-\tau)X(\tau)d\tau \tag{4.14}$$

The time t may or may not be in the interval $[a, b]$. $X(t)$ represents the measured data. $h(\tau)$ is treated as a set of weighting functions. The error should be *orthogonal* to the data. Thus,

$$E[(Y(t)-\hat{Y}(t))X(s)] = E\left[Y(t)X(s) - \int_a^b h(t-\tau)X(\tau)X(s)d\tau\right] = 0 \tag{4.15}$$

This implies that

$$E[Y(t)X(s)] = E\left[\int_a^b h(\tau)X(t-\tau)X(s)d\tau\right] \tag{4.16}$$

or

$$R_{YX}(t,s) = \int_a^b h(t-\tau)R_X(\tau,s)d\tau \tag{4.17}$$

Eq. (4.17) is known as the *Wiener-Hopf* equation [Ref. 14.]. Theoretically, this result can be used to solve for the weighting function given the assumption that the auto- and crosscorrelation functions involved are known. However, there is no general solution method for all practical applications. Usually, specialized forms based upon one or more simplifying assumptions are solved.

Weiner filters may be *causal* or *noncausal.* The weighting function of a noncausal filter requires the filter to "look ahead" of real-time and use data which is not yet available. The estimate of the output at a particular time *t*, for a noncausal filter is given by

$$\hat{Y}(t) = \int_{-\infty}^{\infty} h(t-\tau)X(\tau)d\tau \tag{4.18}$$

The auto- and crosscorrelation relations can be expressed as

$$R_{YX}(S) = \int_{-\infty}^{\infty} h(\tau)R_X(S-\tau)d\tau \tag{4.19}$$

If is assumed that the processes involved are WSS, a closed form solution for the weighting function can be found in the frequency domain. Taking the Fourier transform of both sides of Eq. (4.19) produces

$$S_{YX}(j\omega) = H(j\omega)S_X(j\omega) \tag{4.20}$$

Thus, by rearranging Eq. (4.20) [Ref. 14.]

$$H(j\omega) = \frac{S_{YX}(j\omega)}{S_X(j\omega)} \tag{4.21}$$

If it is assumed that the input measurement has the following form

$$X(t) = Y(t) + n(t) \tag{4.22}$$

where *n*(*t*) is uncorellated Guassian noise. Then Eq. (4.21) will become [Ref. 86.]

$$H(j\omega) = \frac{S_{YY}(j\omega)}{S_{YY}(j\omega) + S_{nn}(j\omega)} \tag{4.23}$$

Noncausal filters are applicable to applications in which post-processing of data is performed, but are not useful in real-time tracking applications.

Casual systems are dependent only upon the past and present values of input and are therefore applicable in real-time applications. The estimate of the output, for a causal filter is given by [Ref. 14.]

$$\hat{Y}(t+\lambda) = \int_{-\infty}^{0} h_{\lambda}(\alpha)X(t+\alpha)d\alpha \tag{4.24}$$

where t is the "present" time. Unfortunately, there is no closed form solution for the weighting function in Eq. (4.24). Application of methods such as *inovation* and *spectral factorization* become necessary [Ref. 14.].

Application of continuous filter theory to digital computers which are processing sampled data can be difficult. Discretization of a transfer function of a filter formulated in continuous time may not produce the results desired.

2. Discrete Weiner Filters

Wiener filtering of discrete data is also a weighting function approach. The weighting function again attempts to weigh all past data in a manner which produces the best estimate. Given n noisy input measurements at times t_1 through t_n, the estimation becomes

$$\hat{Y} = a_1X(t_1) + a_2X(t_2) + \ldots + a_nX(t_n) \tag{4.25}$$

and the mean square error becomes

$$e = E\{[Y(t) - (a_1X_1 + a_2X_2 + \ldots + a_nX_n)]^2\} \tag{4.26}$$

To find the minimum of the squared error criterion in Eq. (4.26), the partial derivative with respect to each a_i is taken.

$$\begin{aligned}\frac{\partial e}{\partial a_i} &= E[(Y(t) - \hat{Y}(t))X(t_i)] \\ &= E[Y - (a_1X_1 + a_2X_2 + \ldots + a_nX_n)X_i] \\ &= E[YX_i - a_1X_iX_1 - a_2X_iX_2 - \ldots - a_nX_iX_n] \\ &= E[YX_i] - a_1E[X_iX_1] - a_2E[X_iX_2] - \ldots - a_nE[X_iX_n] \\ &= R_{XY}(t, t_i) - a_1R_X(t_i, t_1) - a_2R_X(t_i, t_2) - \ldots - a_nR_X(t_i, t_n) = 0\end{aligned} \tag{4.27}$$

These n resulting equations can be expressed in matrix form by [Ref. 14.]

$$Ra = \begin{bmatrix} R_X(t_1, t_1) & \dots & R_X(t_n, t_1) \\ \cdot & & \cdot \\ \cdot & & \cdot \\ R_X(t_1, t_i) & \dots & R_X(t_n, t_i) \\ \cdot & & \cdot \\ \cdot & & \cdot \\ R_X(t_1, t_n) & \dots & R_X(t_n, t_n) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \cdot \\ \cdot \\ \cdot \\ a_n \end{bmatrix} = \begin{bmatrix} R_{XY}(t_1, t) \\ \cdot \\ \cdot \\ \cdot \\ R_{XY}(t_n, t) \end{bmatrix} = R_0 \tag{4.28}$$

This implies that a can be solved for by inverting the n x n matrix R.

$$a = R^{-1}R_0 \tag{4.29}$$

The above assumes that each of the auto and crosscorrelation functions in Eq. (4.28) is known.

Inversion of R in Eq. (4.29) can be computationally expensive. This inversion must be completed each time a new estimate is required. As the size of the data set increases with time, the growing dimension of R will soon make the problem intractable. A limitation may be placed upon the number of previous measures used, but inversion of an n x n matrix will still be necessary each time a new data point is received. It should also be noted the Eq. (4.28) takes into account only one input and one output. If multiple outputs are involved, there will be multiple matrices to be inverted.

F. KALMAN FILTERING

The Kalman filter is an alternate method of formulating the linear minimum mean-square error filtering problem which utilizes state space methods [Ref. 14.]. The two main features of the Kalman formulation of the problem are vector modeling of the random processes under consideration and recursive processing of the noisy measurement data vector. Unlike the discrete time Wiener filter which must reprocess all previous data each time a new estimate in required, recursive processing allows an updated estimate to be made using only the results from the previous estimate.

Kalman filter theory continues the assumption that the spectral characteristics of the processes involved are known. All noise sources are assumed to be *white* and *Gaussian*

[Ref. 49.]. "Whiteness" implies that the noise values are not correlated in time and have equal power in all frequencies. Gaussian noise amplitude takes on the shape of a normal-bell shaped curve. The probability density of a Gaussian noise source is completely described by its mean and variance. Under these assumptions a Kalman filter will produce an *optimal estimate* of the variables of interest. This optimality is based on Bayes theorem and the use of conditional probability density functions [Ref. 91.]. Continuous Kalman filters are only of theoretical interest and are rarely used in practical applications and thus will not be discussed here.

1. Discrete Kalman Filters

Discrete Kalman filter theory is primarily based upon a *process model* and the *measurement equation.* The process model express the physical characteristics of the system. It predicts how the state of the system changes from one time step to the next. Through the process model, unreasonable estimates made using only sensor data may be discounted. This model for change is written [Ref. 91.]

$$X_{n+1} = \Phi_n X_n + W_n \tag{4.30}$$

where

- X_{n+1} and X_n are n x 1 state vectors expressing the state of the system at the times n +1 and n respectively.
- Φ_n is an n x n constant state transition matrix expressing the physical equations which govern system state transitions.
- W_n is a n x 1 process noise vector. The n independent white noise sources have a known covariance and account for system inaccuracies.

The measurement equation [Ref. 91.]

$$Z_{n+1} = H_n X_n + V_n \tag{4.31}$$

expresses how measurement data is related to the state of the system. Based on a given set of measurements, it defines what state the system should be in. Individual terms are as follows

- Z_n is an m x 1 vector of measurement data at time n.

- X_n is an n x 1 state vector expressing the state of the system.
- H_n is an m x n constant measurement matrix which relates measurements to the system state.
- V_n is an n x 1 measurement noise vector. The n independent white noise sources account for measurement inaccuracies. These noise sources are uncorellated with W_n in Eq. (4.30) and have a known covariance.

The covariance matrix for the process noise vector, W_n, and the measurement noise vector, V_n, vectors is given by

$$Q_k = E[W_k W_k^T] = E[W_k^2] \tag{4.32}$$

and

$$R_k = E[V_k V_k^T] = E[V_k^2] \tag{4.33}$$

The estimation error is expressed as

$$e_{\bar{k}} = X_k - \hat{X}_{\bar{k}} \tag{4.34}$$

and the associated error covariance matrix is

$$P_{\bar{k}} = E[e_{\bar{k}} e_{\bar{k}}^T] = E[(X_k - \hat{X}_{\bar{k}})(X_k - \hat{X}_{\bar{k}})^T] \tag{4.35}$$

where the super-minus indicates that the best estimate prior to assimilating the actual measurement at the corresponding time. The discrete linear estimation is

$$\hat{X}_k = (I - K_k H_k)\hat{X}_{\bar{k}} + K_k Z_k \tag{4.36}$$

For clarity, Eq. (4.36) can be rearranged and written as

$$\hat{X}_k = \hat{X}_{\bar{k}} + K_k(Z_k - H_k \hat{X}_{\bar{k}}) \tag{4.37}$$

The second term on the right side of the equation expresses the error or update. The subtraction in the term produces the difference between the actual measurement and the expected measurement. The n x n weighting matrix, K_k, is the *Kalman gain matrix*, which is given by [Ref. 14.]

$$K_k = P_{\bar{k}} H_k^T (H_k P_{\bar{k}} H_k^T + R_k)^{-1} \tag{4.38}$$

Evaluation of Eq. (4.38) requires inversion of an n x n matrix.

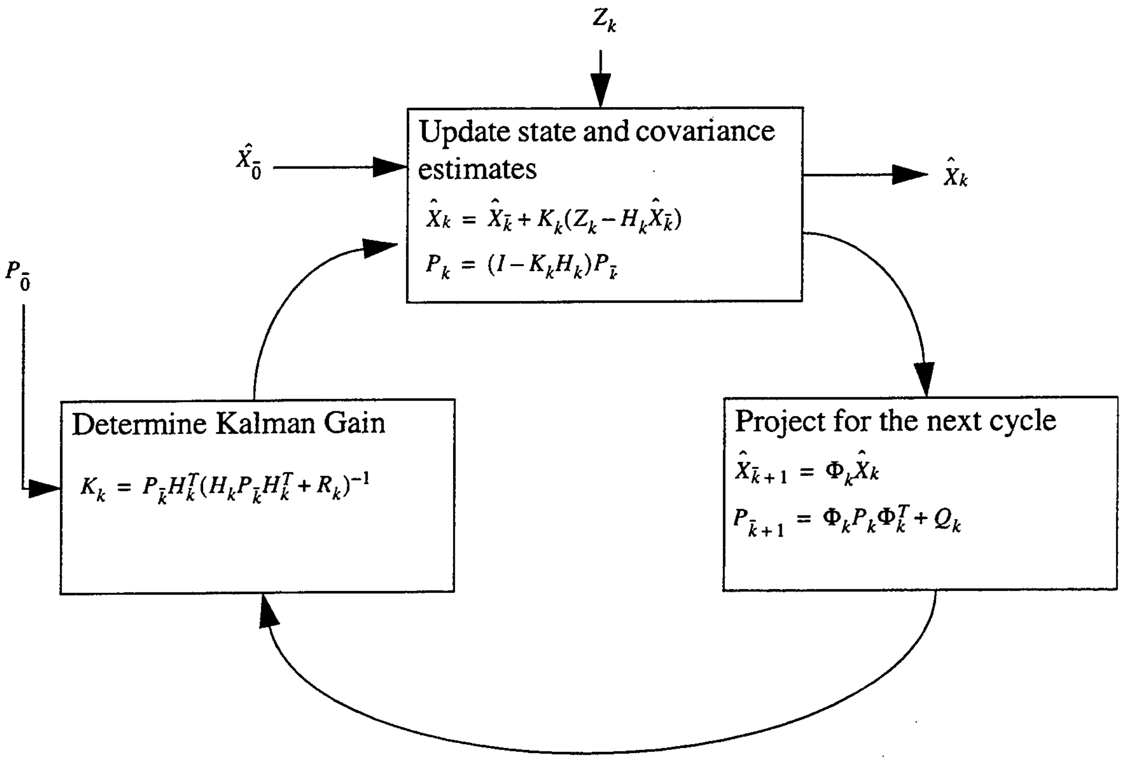

Z_k
$\hat{X}_{\bar{0}}$
Update state and covariance estimates
$\hat{X}_k = \hat{X}_{\bar{k}} + K_k(Z_k - H_k\hat{X}_{\bar{k}})$
$P_k = (I - K_kH_k)P_{\bar{k}}$
$\hat{X}_k$
$P_{\bar{0}}$
Determine Kalman Gain
$K_k = P_{\bar{k}}H_k^T(H_kP_{\bar{k}}H_k^T + R_k)^{-1}$
Project for the next cycle
$\hat{X}_{\bar{k}+1} = \Phi_k\hat{X}_k$
$P_{\bar{k}+1} = \Phi_kP_k\Phi_k^T + Q_k$

Figure 7: Kalman Filter Loop After [Ref. 14.]

Once initial estimates of the state ($\hat{x}_{\bar{k}}$), and the error covariance ($P_{\bar{k}}$) are determined, Eq. (4.30) and Eq. (4.31) are repeatedly used to obtained updated estimates of the system state as depicted in Figure 7. The elements of Kalman gain matrix will continue to change during operation. Examination of Eq. (4.38) reveals that the only non-constant term is the error covariance matrix. Thus, changes in the filter gain are directly related to the estimated accuracy of the current state estimate. In effect, a Kalman filter automatically provides information about the quality of the estimates while doing the estimation through $P_{\bar{k}}$.

2. Extended and Linearized Kalman Filters

In some applications, either the dynamic or measurement relations may be nonlinear. The measurement equation may be a nonlinear function of the state variables, the process model may be nonlinear function of the state variables, or both. These relations can be expressed as

$$X_{n+1} = g(X_n) + W_n \tag{4.39}$$

and

$$Z_{n+1} = f(X_n) + V_n \tag{4.40}$$

where f and g are nonlinear functions of the state.

Eq. (4.39) and Eq. (4.40) may be linearized by taking partial derivatives of the nonlinear functions. Under such conditions the transition or measurement matrices will no longer be constant and must be updated each time a new estimate of the state is made. There are two basic methods of linearization. In a *linearized* Kalman filter, the partial derivative of g or f is taken with respect to some nominal trajectory which does not involve the measurement data [Ref. 14.]. In an *extended* Kalman filter, the partial derivative of g or f may be taken with respect to the current state estimate [Ref. 91.]. The resulting matrix of first partial derivatives or the *Jocobian* is given by

$$H_n = \frac{\partial f}{\partial x} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} \frac{\partial f_1}{\partial x_2} \cdots \\ \frac{\partial f_2}{\partial x_1} \frac{\partial f_2}{\partial x_2} \cdots \\ \cdots \end{bmatrix} \qquad \Phi_n = \frac{\partial g}{\partial x} = \begin{bmatrix} \frac{\partial g_1}{\partial x_1} \frac{\partial g_1}{\partial x_2} \cdots \\ \frac{\partial g_2}{\partial x_1} \frac{\partial g_2}{\partial x_2} \cdots \\ \cdots \end{bmatrix} \tag{4.41}$$

In either case the actual filter remains linear and performs its estimation using a linearized model or approximation of a nonlinear problem.

Neither method of linearization is without risks. Linearized and extended Kalman filters can no longer be proved to be optimal based on a derivation using Bayes theorem [Ref. 91.]. In an extended Kalman filter, there is a potential for bad estimates to get worse and lead to an eventual divergence of the filter. This may be especially true under circumstances in which the initial uncertainty and measurement errors are large. Linearized Kalman filters will be inaccurate in situations in which the nominal trajectory does not closely match the actual trajectory. Recognition and correction of poor performance becomes a key component in the design of such filters.

The *shock* of a Kalman filter expresses the difference between what is actually measured and the best prediction of the state. The shock for extended Kalman filter at time *i* this can be expressed as

$$S_i = Z_i - f(\mu_X) - \mu_V \tag{4.42}$$

where μ_X is the mean of the system state and μ_V is the mean of the measurement noise. The magnitude of a *dimensionless shock* term is given by

$$DS_i = S_i^T(H_i P_i(-H_i^T) + R_i)S_i \tag{4.43}$$

Should the magnitude of Eq. (4.43) become large compared to the number of components of S_i, it is likely that the filter has lost track [Ref. 91.].

Extended and linearized Kalman filters have performed well in a variety of applications. However, it must be recognized that the added complexity of these types of filters makes them more computationally demanding than other types of filters. Recalculation of the Jacobian during each update cycle takes time. The complexities of the nonlinear models involved may make it difficult to produce updated state estimates in a timely manner.

G. COMPLEMENTARY FILTERING

Both Weiner and Kalman filter theory are based on the assumption that the spectral characteristics of the processes involved are known. In practical applications this assumption is often difficult to satisfy. It may be impractical to model the input signal as a random process with known spectral characteristics. Complementary filters are "ad-hoc" systems which are not dependent upon these strict assumptions. Though Weiner or Kalman filter theory may be used to select an appropriate transfer function, neither method is required. Complementary filters filter the input signal without unnecessary delay or distortion.[Ref. 14.]

Complementary filtering is based upon the use and availability of multiple independent noisy measurements of the same signal. If the measurements have complementary spectral characteristics, transfer functions may be chosen in such a way as to minimize estimation error. The general requirement is that one of the transfer functions complement the sum of the others. Thus, for n measurements of a signal [Ref. 14.]

$$1 - H_1(s) - H_2(s) - \ldots - H_{n-1}(s) = H_n(s) \qquad (4.44)$$

This will allow the signal component to pass through the system undistorted since the output of the system will always sum to one. The simplest complementary filter involves two noise contaminated measurements of a signal. This situation is depicted in Figure 8. If N_1 is predominantly low-frequency noise and N_2 is high frequency noise, the two noise sources have complementary spectral characteristics. Choosing *H*(*s*) to be a low-pass filter attenuates both noise signals. The output can be written [Ref. 14.]

$$Y(s) = X(s) + N_1(s)[1 - H(s)] + N_2(s)H(s) \qquad (4.45)$$

where

$$H(s) = \frac{1}{1 + \tau s} \qquad (4.46)$$

which satisfies the conditions required by Eq. (4.44). Since both high and low frequency data are utilized, the filter output will not suffer from any delay in dynamic response due to low-pass filtering.

Examination of Eq. (4.45) indicates that the filter only operates upon the errors and noise involved in the system. The transfer function does not directly affect the input signal itself. For this reason, this type of filtering is sometimes called *distortionless filtering*. [Ref. 14.]

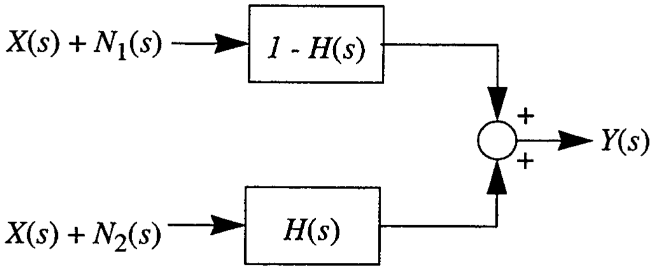


Figure 8: Complementary Filter Block Diagram

Figure 9 depicts a constant gain complementary filter for attitude estimation. The transform of the roll angle from accelerometer readings due to gravity is $\varphi_a(s)$, while $\varphi_s(s)$ is the roll angle obtained by integrating rate signals. If the accelerometer were noiseless and sensed only gravitational acceleration, *k* would be set to infinity and the attitude estimation would be entirely accelerometer based. Use of noiseless rate sensors with no bias would allow attitude estimation using only these sensors and *k* could be set to zero. Since neither sensor is ideal, a compromise value for *k* that gives the best estimate must be found.

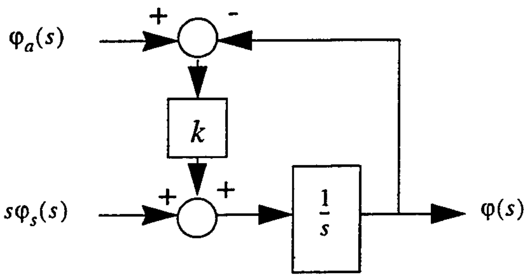


Figure 9: Transform Domain Block Diagram Of Roll Angle Estimation Filter

From Figure 9, the output of the filter is given by

$$\varphi(s) = \frac{1}{s}(k\varphi_a(s)) - k\varphi(s) + s\varphi_s(s) \tag{4.47}$$

The filter transfer function based on accelerometer input alone with $\varphi_s(s) = 0$ is given by

$$G_a(s) = \frac{\varphi(s)}{\varphi_a(s)} = \frac{ks^{-1}}{1+ks^{-1}} = \frac{k}{s+k} = \frac{1}{1+\tau s} \tag{4.48}$$

where $\tau = \frac{1}{k}$. With a unit step input, *u*(*t*), the frequency domain output of the filter is

$$\varphi_a(s) = \frac{1}{s}\left(\frac{k}{s+k}\right) = \frac{k}{s(s+k)} = \frac{1}{s} - \frac{1}{s+k} \tag{4.49}$$

The far right expression is derived through partial fraction expansion [Ref. 41.]. Transforming to the time domain produces

$$\varphi_a(t) = 1 - e^{-kt} = 1 - e^{-\frac{t}{\tau}} \tag{4.50}$$

Since $e^{-1} \approx 0.37$, when *t* equals τ, the filter output due to accelerometer input has increased to $1 - 0.37 = 0.63$ or 63 percent of its steady state value. Therefore, the accelerometer input is low pass filtered.

Similarly, the transfer function for rate sensor input alone with $\varphi_a(s) = 0$ is

$$G_s(s) = \frac{\varphi(s)}{\varphi_s(s)} = \frac{1}{1+ks^{-1}} = \frac{s}{s+k} = \frac{\tau s}{1+\tau s} \tag{4.51}$$

With a unit step input, the frequency domain rate sensor output of the filter is

$$\varphi_s(s) = \frac{1}{s}\left(\frac{s}{s+k}\right) = \frac{1}{s+k} = \frac{\tau s}{s\tau+1} \tag{4.52}$$

Transforming to the time domain produces

$$\varphi_s(t) = e^{-kt} = e^{-\frac{t}{\tau}} \tag{4.53}$$

In this case when $t = \tau$, the output due to rate sensor input will have decreased to 37 percent of its initial value. Eq. (4.53) high pass filters the rate sensor data.

From Eq. (4.48) and Eq. (4.51), the combined transfer function due to both rate sensor and accelerometer input is

$$\frac{\varphi(s)}{\varphi_a(s)} + \frac{\varphi(s)}{\varphi_s(s)} = \frac{1}{1+\tau s} + \frac{\tau s}{1+\tau s} = \frac{1+\tau s}{1+\tau s} = 1 \tag{4.54}$$

which sums to unity regardless of the value of k [Ref. 56.]. Transforming the sum into the time domain produces the total response of the filter

$$\left(1-e^{-\frac{t}{\tau}}\right)+e^{-\frac{t}{\tau}} = 1 \tag{4.55}$$

This means that the initial response of the filter to a step change comes entirely from rate input. The rate input decays exponentially over time and is replaced by complementary "low frequency" accelerometer input.[Ref. 56.]

1. Crossover Frequency

The crossover frequency of a complementary filter represents the value below which signals from one type of sensor are given a greater weight and above which signals from another type of sensor are favored. At the crossover frequency, signals from both inputs are weighted equally. For the filter depicted in Figure 9, below the crossover frequency accelerometer signals are given greater weight. Above, the rate sensor signals are more trusted.

The crossover frequency of a filter of the form in Figure 9 can be found by equating the absolute values of the separate transfer functions [Ref. 50.]. Rewriting in the complex frequency domain and equating the transfer functions from Eq. (4.48) and Eq. (4.51) produces

$$\left|\frac{\varphi(j\omega)}{\varphi_a(j\omega)}\right| = \left|\frac{\varphi(j\omega)}{\varphi_s(j\omega)}\right| \tag{4.56}$$

The magnitudes of the transfer functions are given by

$$\left|\frac{\varphi(j\omega)}{\varphi_a(j\omega)}\right| = \left|\frac{1}{1+j\tau\omega}\right| = \frac{1}{\sqrt{1+\tau^2\omega^2}} \tag{4.57}$$

and

$$\left|\frac{\varphi(j\omega)}{\varphi_s(j\omega)}\right| = \left|\frac{j\tau\omega}{1+j\tau\omega}\right| = \frac{\tau\omega}{\sqrt{1+\tau^2\omega^2}} \tag{4.58}$$

Thus, at the crossover frequency

$$1 = \tau\omega \tag{4.59}$$

which implies

$$\omega = \frac{1}{\tau} \tag{4.60}$$

In Hertz, the crossover frequency, f_c, can be written as

$$f_c = \frac{1}{2\pi\tau} = \frac{k}{2\pi} \tag{4.61}$$

and can be adjusted by varying the filter gain k.[Ref. 51.]

H. SUMMARY AND CONCLUSIONS

Each of the types of filters reviewed above has its own sets of strengths and weaknesses. They differ in computational complexity, memory requirements, and applicability to discrete implementation on digital computers. They also differ in the assumptions on which the underlying theory is based and applicability to problems involving numerous variables to be estimated and multiple outputs. However, the primary goal of each is to produce the best possible estimate for the variable or variables of interest by minimizing errors due to noise corrupted measurements and inaccuracies due to sensor limitations and the precision of the system.

Weiner filter theory is applicable to filtering problems involving the separation of one noiselike signal from another. The end result of solving an integral equation is a weighting function which describes the relationship between input and output. Weiner filter theory is completely based upon the assumption that spectral characteristics of both the signal and noise are known and uses only this information to minimize the mean square error. However, in many practical applications the auto and crosscorrelation functions may not be known. The scalar formulation of Weiner filter theory makes it difficult to apply to problems involving multiple inputs and outputs. Though there may be multiple inputs, only a single scalar output may be estimated. Perhaps the greatest obstacle to the discrete implementation of a Wiener filter on a digital computer is the requirement that the solution be completely recalculated each time additional data in obtained. This requires that all previous measurement data be stored in memory and be available for recalculation of the solution. As the size of the data set grows, inversion of the covariance matrix soon becomes intractable.

Kalman filter theory, like Weiner filter theory assumes that the spectral characteristics of each signal is completely specified. While Weiner filters use constant gains, Kalman filters have time varying gains which are derived using the Kalman gain matrix. The Kalman filter incorporates a physical process model as part of the estimation process. The end result is a differential or difference equation relating input and output. The matrix formulation of the Kalman filter makes it applicable to a large class of problems involving multiple inputs and outputs as well as complex measurement and process relationships. Discrete Kalman filters are particularly applicable to implementation on a computer due to their recursive nature. It is not required that all previous data to be kept in storage and reprocessed every time a new measurement is taken. Only the most recent estimate and measurement are needed to arrive at a new estimate of the state of the system. Kalman filter theory does assume that all noise sources are white and Gaussian. However, it can be proved that the sum of multiple colored noise sources will result in a Gaussian distribution and thus a Kalman filter will still perform well even when the assumptions are not true [Ref. 49.]. The traditional Kalman filter is based upon a linear process model and measurement equation. Though the filter can no longer be proved to be optimal, a nonlinear process model can be used in a linearized or extended Kalman filter. Formulation of such a process model can be extremely difficult and time consuming. Other difficulties can arise due to the additional computational demands of linearization.

Kalman filters are highly reliant on having complete measurement statistics and an accurate process model. In the absence of either of these requirements, highly inaccurate estimates of the system state can result [Ref. 14.]. Complementary Filters are not based upon the assumption of having complete statistical data regarding the signals involved in the problem and thus are often more robust. Most commonly they are designed to combine multiple measurements of the same signal in a complementary fashion. The primary goal continues to be minimization of the square of the expected error. Any appropriate parameter optimization technique can be used to solve the minimization problem. Often, a complementary filter is tuned using empirical data obtained in experimental trials of the system. The formulation of a complementary filter is usually more straightforward and simpler than that of a Kalman filter. Though not optimal, a complementary filter can produce estimates with an accuracy which is comparable to that of an Kalman filter, with a lower computational overhead and less development time.

It was stated at the beginning of this chapter that inertial/magnetic tracking of human body segments is basically a navigation problem. In recent years, this type of problem has most commonly been solved using a complementary filter to integrate the data from multiple complementary sensors. Foxlin has had success using a reduced order extended Kalman filter in similar but simpler head tracking applications in which inertial sensors were used [Ref. 27.]. The ideal solution to the body tracking problem would be an extended Kalman filter which incorporates a dynamic model of the human musculoskeletal system, and measurement statistics of the sensors. Dynamic models for the musculoskeletal system have been studied for many years [Ref. 23.]. Such models are ideal for computer simulations of articulated body motions, but they are currently too computationally demanding for real-time applications such as human motion tracking. Thus, the challenge would be to develop a model that is adequate, but not overwhelmingly complex for motion tracking applications. In the end however, it may be the case that a properly tuned complementary filter will provide estimates with an accuracy that is comparable to those made by an extended Kalman filter without the associated complexity and development time. Thus, the prototype research described here makes use of a complementary filter based upon a quaternion representation of orientation and leaves the development of an extended Kalman filter for this application to future work [Ref. 48.].

V. A QUATERNION ATTITUDE FILTER

A. INTRODUCTION

Human body tracking using inertial sensors requires an attitude estimation filter capable of tracking in all orientations. Singularities associated with Euler angles make them unsuitable for use in body tracking applications. Quaternions provide an alternate method of orientation representation that is more efficient than the use of rotation matrices and does not involve the use of trigometric functions. In addition, quaternions do not suffer from the singularities associated with Euler angles.

The optimality of Kalman filter theory is entirely based upon the assumption that complete statistical data regarding the signals involved in the problem are known. In practice this may not be true. Calculation of the Kalman gains requires the inversion of an *n* x *n* matrix on each iteration step. In a nonlinear problem such as human-body tracking, it becomes necessary to use an extended Kalman filter. In this case it may be necessary to compute Jocobians to linearize both the measurement and process model equations at each iteration step. In order to keep the problem tractable, it may also be necessary to simplify the involved process model to the point where it is no longer accurate.

Nonlinear regression analysis is a simpler form of optimal least-squares estimation. In this method, a *squared error criterion function* relating the measurements to the state estimate is minimized using a *least squares estimate* of the true value of the state. The least squares estimate can be derived using techniques such as Gauss-Newton and Newton iteration. This chapter describes the theory, design, and analysis of a complementary attitude estimation filter based upon a quaternion representation of orientation and Gauss-Newton iteration.

B. A QUATERNION ATTITUDE FILTER

Figure 10 is a block diagram of the complementary quaternion-based attitude estimation filter used in this research. The filter takes inputs from three separate sensors.

Its output is a unit quaternion representation of the orientation of the tracked object, $\hat{q}$. The inputs are from a three-axis angular rate sensor (p, q, r), a three-axis accelerometer (h_1, h_2, h_3), and a three-axis magnetometer (b_1, b_2, b_3).

In an error free, noiseless world, angular rate data could be processed to obtain a rate quaternion using the relationship

$$\dot{q} = q\left(0,\ \frac{1}{2}p\ \frac{1}{2}q\ \frac{1}{2}r\right) = \frac{1}{2}q\ {}^{B}\omega \tag{5.1}$$

where the indicated product is a quaternion product and the superscript ^{B} means measured in body coordinates (See Chapter III for a complete derivation of Eq. (5.1)). Single integration of $\dot{q}$ would produce a quaternion which describes orientation. However, in an environment containing noise and errors, the output of angular rate sensors would tend to drift over time. Thus, rate sensor data can be used to determine orientation only for relatively short periods of time unless this orientation is continuously corrected using "complementary" data from additional sensors.

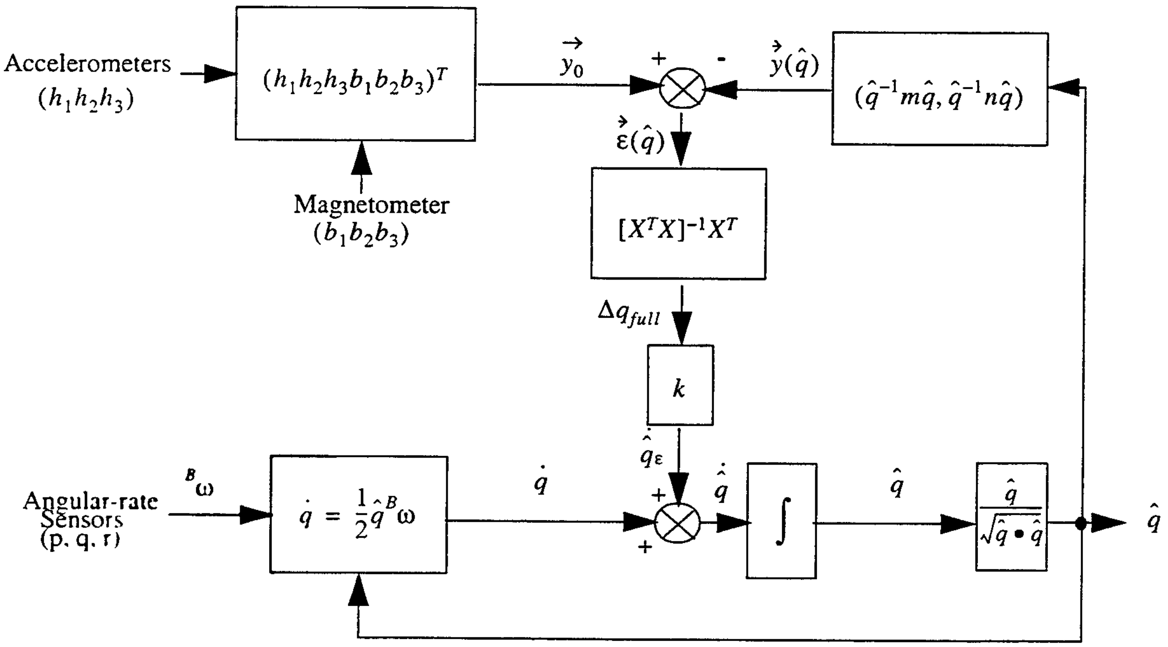

Accelerometers
$(h_1 h_2 h_3)$
$(h_1 h_2 h_3 b_1 b_2 b_3)^T$
Magnetometer
$(b_1 b_2 b_3)$
$\vec{y}_0$
+
-
$\vec{y}(\hat{q})$
$(\hat{q}^{-1} m\hat{q}, \hat{q}^{-1} n\hat{q})$
$\vec{\varepsilon}(\hat{q})$
$[X^T X]^{-1} X^T$
Δq_{full}
k
$\dot{\hat{q}}_\varepsilon$
Angular-rate
Sensors
(p, q, r)
$^B\omega$
$\dot{q} = \frac{1}{2}\hat{q}\,{}^B\omega$
$\dot{q}$
+
+
$\dot{\hat{q}}$
$\int$
$\hat{q}$
$\frac{\hat{q}}{\sqrt{\hat{q} \bullet \hat{q}}}$
$\hat{q}$

Figure 10: Quaternion-Based Attitude Filter From [Ref. 8.]

Accelerometers measure the combination of forced linear acceleration and the reaction force due to gravity. That is,

$$\vec{a}_{measured} = \vec{a} - \vec{g} \tag{5.2}$$

Since most real-life objects do not experience constant linear acceleration, when averaged over time, accelerometers return a gravity vector or the local vertical. Thus, accelerometer outputs are used to correct orientation relative to a vertical axis. Similarly, magnetometers measure the local magnetic field in body coordinates. This information is used to correct rate sensor drift errors in the horizontal plane.

1. Parameter Optimization

Combining filter inputs can be regarded as a parameter optimization problem with the goal of minimizing modeling error. The closer the estimated orientation to the actual orientation, the smaller the modeling error. Through iteration and calculations based on the magnitude and direction of modeling errors, orientation estimations become increasingly accurate. Theoretically, when the modeling error is zero, the estimated orientation is equal to the actual orientation.

The three orthogonally mounted accelerometers return an approximation to the local vertical, the unit vector *h.* The magnetometer returns the direction of the local magnetic field, *b,* also normalized to a unit vector. These two vector quantities expressed in body coordinates as pure imaginary unit quaternions are

$$h = [0 \;\; h_1 \;\; h_2 \;\; h_3] \qquad b = [0 \;\; b_1 \;\; b_2 \;\; b_3] \tag{5.3}$$

Combining the vector parts of $Ve(h)$ and $Ve(b)$ from Eq. (5.3) produces a 6 x 1 *measurement vector* representing the actual measurements taken by the accelerometers and magnetometers.

$$\vec{y}_0 = [Ve(h), Ve(b)]^T = [h_1 h_2 h_3 b_1 b_2 b_3]^T \tag{5.4}$$

Gravity in earth coordinates is always down and can be expressed as the down unit vector in quaternion form as

$$m = [0\ \ 0\ \ 0\ \ 1] \tag{5.5}$$

The local magnetic field in earth coordinates, once determined and normalized, can be expressed in unit quaternion form as

$$n = [0\ \ n_1\ n_2\ n_3] \tag{5.6}$$

Eq. (5.5) and Eq. (5.6) are transformed from earth fixed coordinates to body coordinates through quaternion multiplication with the estimated orientation, $\hat{q}$ by [Ref. 92.]

$$h = \hat{q}^{-1}m\hat{q} \qquad b = \hat{q}^{-1}n\hat{q} \tag{5.7}$$

Combining the imaginary parts of Eq. (5.7) into a single 6 x 1 *computed measurement vector* produces

$$\vec{y}(\hat{q}) = [Ve(\hat{q}^{-1}m\hat{q}), Ve(\hat{q}^{-1}n\hat{q})]^T = [\hat{h}_1\hat{h}_2\hat{h}_3\hat{b}_1\hat{b}_2\hat{b}_3]^T \tag{5.8}$$

Eq. (5.4) represents the measured gravity vector and local magnetic field while Eq. (5.8) is the computed gravity vector and magnetic field found using Eq. (5.7) and is based upon the best estimate of the current orientation. The difference between the actual measurements and the computed measurement is the *error vector* or *modeling error*

$$\vec{\varepsilon}(\hat{q}) = \vec{y}_0 - \vec{y}(\hat{q}) \tag{5.9}$$

In viewing Eq. (5.9), note that if $\hat{q} = q_{true}$ in Eq. (5.7) and there is no measurement noise, the difference between the measured and computed values, $\vec{\varepsilon}(q)$, will equal the zero vector.

The square of the filter modeling error is termed the *criterion function*

$$\varphi(\hat{q}) = \vec{\varepsilon}^{\,T}(\hat{q})\vec{\varepsilon}(\hat{q}) \tag{5.10}$$

In the current version of the filter, $\varphi(\hat{q})$ is minimized using Gauss-Newton iteration [Ref. 59.]. This method is based on linearized least squares regression analysis where $\vec{y}_0$ is considered a vector of data points and $\vec{y}(\hat{q})$ is a vector to be fitted to those points. The full correction step to the measured rate quaternion is [Ref. 59.]

$$\Delta q_{full} = \left[X^T X\right]^{-1} X^T \varepsilon(\hat{q}) = S^{-1} X^T \varepsilon(\hat{q}) \tag{5.11}$$

where the X matrix is defined as

$$X_{ij} = \left[\frac{\partial y_i}{\partial \hat{q}_j}\right] \tag{5.12}$$

It should be noted that if $\hat{q}$ is not constrained to unit length as depicted in Figure 10 and discussed in Appendix B, a unique solution to the problem it no longer exists and the X matrix will not be of full rank. In this case the regression matrix

$$S = X^T X \tag{5.13}$$

will be singular and can not be inverted. The orthogonal quaternion theorem described later in this chapter provides a method of avoiding regression matrix singularities and improving filter efficiency.

Eq. (5.11) treats m and n as if they are perfect measurements of forced linear acceleration and the local magnetic field. In dealing with data corrupted by noise, a scalar multiplier α is used.

$$\Delta q_{partial} = \alpha \left[X^T X\right]^{-1} X^T \vec{\varepsilon}(\hat{q}) \tag{5.14}$$

where $0 < \alpha < 1$. In the absence of noise, α would be set to nearly unity. Very noisy or inaccurate measurements would demand that the scalar multiplier α be given a value closer to zero. For a discrete approximation to a continuous time filter, referring to Figure 10

$$\alpha = k\Delta t \tag{5.15}$$

Thus, for discrete time step integration, the next estimate of orientation would be

$$\hat{q}_{n+1} = \hat{q}_n + \frac{1}{2}\hat{q}_n {}^B\omega\Delta t + \alpha[X^T X]^{-1} X^T \varepsilon(\hat{q}_n) = \hat{q}_n + k\Delta t \Delta q_{full} + \dot{q}_{measured}\Delta t \tag{5.16}$$

In the continuous time domain, Eq. (5.16) becomes

$$\dot{\hat{q}} = \dot{\hat{q}}_\varepsilon + \dot{q}_{measured} = k\Delta q_{full} + \dot{q}_{measured} \tag{5.17}$$

2. Analysis

Figure 11 is a time domain signal flow graph (SFG)[Ref. 41.] of the linearized quaternion attitude estimation filter. The inputs n_1 and n_2 are maneuver induced noise and rate sensor noise respectively. The basis for linearization is the assumption that in the absence of measurement noise the computation of Δq_{full} is exact and therefore

$$\Delta q_{full} = q_{true} - \hat{q} \tag{5.18}$$

This assumption would be correct only if y depended linearly on q, which it does not. Nevertheless, simulation studies [Ref. 51.] and physical experiments show that this equation offers a very useful approximation for the selection of filter gains and predication of filter response.

Application of Mason's formula [Ref. 41.] to Figure 11 produces

$$\frac{\hat{q}}{\dot{q}_{true}} = \frac{kp^{-2}+p^{-1}}{1+kp^{-1}} = \frac{p^{-1}(1+kp^{-1})}{1+kp^{-1}} = p^{-1} \tag{5.19}$$

where p^{-1} is the time integration operator [Ref. 41.]. Thus, with correct initial conditions, in the absence of noise,

$$\hat{q} = p^{-1}\dot{q}_{true} = q_{true} \tag{5.20}$$

regardless of the value of k. This means that, under the linearization assumptions, Figure 10 is a complementary filter since, for all k, if n_1 and n_2 are zero, then $\hat{q} = q_{true}$.

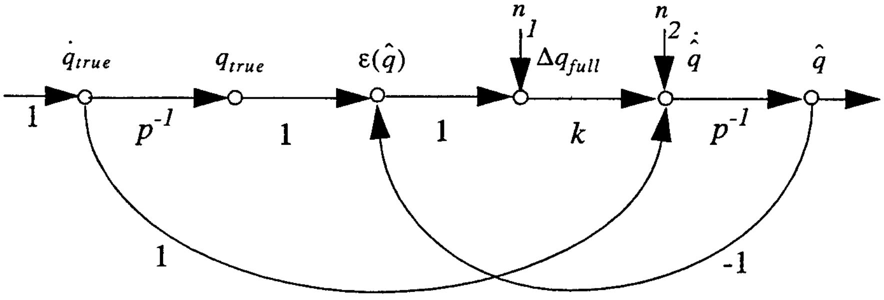


Figure 11: Signal Flow Graph for Linearized System After [Ref. 54.]

a. Noise Response

Applying Mason's formula to noise disturbances n_1 and n_2 in Figure 11 produces the following low pass filter transfer functions.

$$\frac{\hat{q}}{n_1} = \frac{kp^{-1}}{1+kp^{-1}} = \frac{k}{p+k} \tag{5.21}$$

$$\frac{\hat{q}}{n_2} = \frac{p^{-1}}{1+kp^{-1}} = \frac{1}{p+k} \tag{5.22}$$

Eq. (5.21) and Eq. (5.22) can be used to find an optimal k value in Eq. (5.17) based upon power spectral density functions for both the noise signals and actual maneuvering behavior of the tracked object. Unfortunately, this information is typically not available, so ad hoc "tuning" of k must usually performed based upon experimental results. [Ref. 96.]

b. Response to Initial Condition Errors

Eq. (5.20) assumes that $\hat{q}$ has been correctly initialized. In order to understand how an erroneous $\hat{q}$ approaches q_{true} over time, consider the following static sensor scenario. Suppose the sensor is mounted in a static fixture so that all Euler angles are zero and thus

$$q_{true} = (1 \;\; 0 \;\; 0 \;\; 0) \tag{5.23}$$

Assume that the unit quaternion $\hat{q}$ is incorrect and is represented by

$$\hat{q}_0 = (1 \;\; \delta_x \;\; \delta_y \;\; \delta_z) \tag{5.24}$$

where all δ are small quantities. In the absence of motion and noise, $\dot{q}_{true} = 0$ and both n_1 and n_2 equal zero. Therefore, Figure 11 can be simplified to Figure 12 as follows:

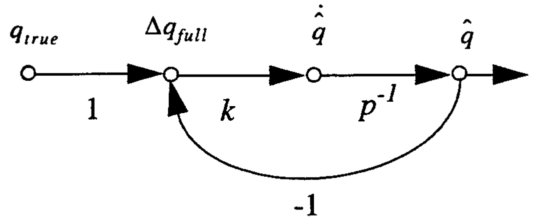


Figure 12: Simplified SFG For Static Testing With Zero Noise After [Ref. 55.]

Based on Figure 12, the initial value for Δq_{full} is

$$\hat{q}_1 = q_{true} - \hat{q}_0 = (0\ -\delta_x\ -\delta_y\ -\delta_z) \tag{5.25}$$

Since the first component of $\hat{q}_1$ in Eq. (5.25) will always be zero, it can be assumed that it will remained unchanged and $\hat{q}$ will take on the form

$$\hat{q} = (1\ \hat{x}\ \hat{y}\ \hat{z}) \tag{5.26}$$

Figure 13 is a Laplace transform SFG for the scalar $\hat{x}$. From the application of Mason's formula it follows that

$$\frac{\hat{X}(s)}{\delta_x} = \frac{s^{-1}}{1 + ks^{-1}} = \frac{1}{s+k} \tag{5.27}$$

Employing the inverse Laplace transform produces the result

$$\hat{x}(t) = \delta_x e^{-kt} \tag{5.28}$$

Equivalent results apply for $\hat{y}(t)$ and $\hat{z}(t)$. This implies that any transient errors in $\hat{q}$ resulting from erroneous initialization will persist for a time inversely proportional to k. Specifically

$$\tau_{\Delta q} = \frac{1}{k} \tag{5.29}$$

and for any disturbance δ_x, the resulting errors in the x component of $\hat{q}$ will be

$$\varepsilon_{\hat{x}}(t) = \delta_x e^{-\tau/\tau_{\Delta q}} \tag{5.30}$$

Thus, it can be predicted that any error will be reduced to 37% of the initial value by the time $t = \tau_{\Delta q}$. Similar results apply to δ_y and δ_z.

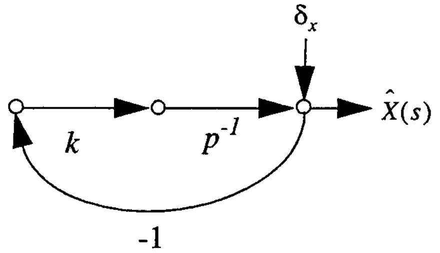


Figure 13: Transform Domain SFG For $\hat{X}(s)$ After [Ref. 55.]

c. *Choosing the Feedback gain value*

If k is too large, the discrete time filter may become unstable or too much maneuver induced error will appear in $\hat{q}$. From Eq. (5.29), it can be seen that k should not be too small if the filter is to converge in a reasonable time period. On the other hand, $\tau_{\Delta q}$ must be larger than the maneuver time constant, $\tau_{maneuver}$, in order to adequately suppress maneuver noise. This result leads to the qualitative requirement

$$\tau_{maneuver} \ll \tau_{\Delta q} \tag{5.31}$$

or

$$1/\tau_{maneuver} \gg k \tag{5.32}$$

The maximum value for k can be quantitatively established through a geometric series [Ref. 7.]. Figure 14 is a block diagram of the linearized quaternion attitude filter. From this diagram, it can be observed that the estimated rate quaternion is given by

$$\dot{\hat{q}} = \dot{q} + \dot{q}_{\varepsilon} \tag{5.33}$$

Discretization of the filter replaces the integral with the summation

$$\int \dot{\hat{q}}\ dt \Rightarrow \sum_{1=0}^{n} \dot{\hat{q}}_{\varepsilon}(n\Delta t)\Delta t \tag{5.34}$$

where

$$\dot{\hat{q}}_{\varepsilon_0} = \dot{\hat{q}}_{\varepsilon}(0), \dot{\hat{q}}_{\varepsilon_1} = \dot{\hat{q}}_{\varepsilon}(1\Delta t), \dot{\hat{q}}_{\varepsilon_2} = \dot{\hat{q}}_{\varepsilon}(2\Delta t), \dots \tag{5.35}$$

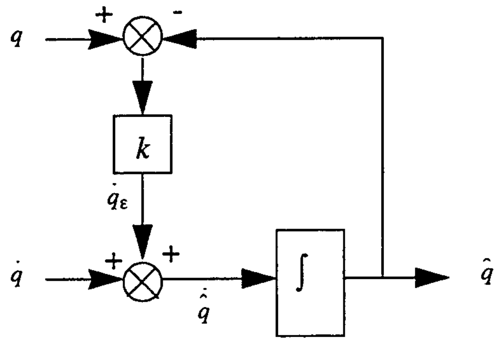


Figure 14: Block Diagram Of Time Domain Linearized Quaternion Attitude Filter

Let $q_{true} = 0$ and assume there is no angular rate input; that is $\dot{q} = 0$. If an error exists then

$$\hat{q}_0 = \hat{q}(0) \neq q_{true} = 0 \tag{5.36}$$

Using Euler integration, the first updated estimate is given by

$$\hat{q}_1 = \hat{q}_0 + \dot{\hat{q}}_0 \Delta t \tag{5.37}$$

Since $\dot{q} = 0$

$$\dot{q}_0 = -k\hat{q}_0 \tag{5.38}$$

Substituting into Eq. (5.37) produces

$$\hat{q}_1 = \hat{q}_0 + (-k\hat{q}_0)\Delta t = (1 - k\Delta t)\hat{q}_0 \tag{5.39}$$

For the second updated estimate

$$\hat{q}_2 = \hat{q}_1 + \dot{\hat{q}}_1 \Delta t \tag{5.40}$$

Again, since $\dot{q} = 0$

$$\dot{q}_1 = -k\hat{q}_1 = -k(1 - k\Delta t)\hat{q}_0 \tag{5.41}$$

Substituting into Eq. (5.40) produces

$$\hat{q}_2 = (1 - k\Delta t)\hat{q}_0 + -k(1 - k\Delta t)\hat{q}_0\Delta t = \ldots = \hat{q}_0(1 - k\Delta t)^2 \tag{5.42}$$

In general, the *nth* estimate is given by the geometric series equation

$$\hat{q}_n = \hat{q}_0(1 - k\Delta t)^n \tag{5.43}$$

Based on this result, it can be observed that for values of $0 \leq k\Delta t < 2$ the geometric series will converge since the absolute value of $1 - k\Delta t$ will be less than unity. The maximum value for *k* for which the filter can expected to be stable is

$$k_{max} = \frac{2}{\Delta t} \tag{5.44}$$

Based on Eq. (5.43) and Eq. (5.14), when working with perfect noiseless data, values for k greater than $\frac{1}{\Delta t}$ can be expected to cause correction "overshoots" and oscillations in the attitude estimate.

The above discussion provides guidelines for the selection of "reasonable" values for k and Δt. With power spectral density functions for $\dot{q}_{true}$, n_1 and n_2, a Kalman filtering approach [Ref. 14.] could be used for this problem. In the absence of such statistical information, gain values may be selected through experimental "tweaking" of the scalar gain, k, in laboratory studies.

3. Reduced Order Filter

The filter derivation discussed above is correct if $\hat{q}$ is constrained to be of unit length. Constraining to unit length also allows formulation of a more efficient algorithm. If it is assumed that the computed measurement vector, $\vec{y}(\hat{q})$, depends linearly on q, the criterion function can be minimized using the relation

$$\hat{q}_{new} = \hat{q}_{old} + \Delta q \tag{5.45}$$

where Δq can be thought of as either a correction to estimated orientation or an update to the old estimate to produce the new estimate. Eq. (5.11) gives the Gauss-Newton iteration formula for Δq as

$$\Delta q_{full} = \left[X^T X\right]^{-1} X^T \varepsilon(\hat{q})$$

Iterative application and recalculation of this correction will lead to convergence for small Δq under known conditions [Ref. 60.].

It should be noted that if $\hat{q}_{old}$ is a positive real unit quaternion, the sum in Eq. (5.45) will not in general be a unit quaternion. However, in order to ensure that a unique solution exists for $\hat{q}$, it should be kept as near to the surface of a four dimensional unit hemisphere as possible. This will be the case if Δq is small and is tangent to the surface of the sphere and thus orthogonal to $\hat{q}$.

Taking the dot product of $\hat{q}$ with itself produces

$$(q+\Delta q)\bullet(q+\Delta q) = q\bullet q+2(q\bullet\Delta q)+\Delta q\bullet\Delta q = 1+2(q\bullet\Delta q)+O(\Delta q^2) \qquad (5.46)$$

If Δq is orthogonal to q, the middle term on the right hand side is zero. The square of the length of $\hat{q}$ is thus given by

$$|q+\Delta q|^2 = 1+O(\Delta q^2) \qquad (5.47)$$

which varies from unity by an order Δq^2 term.

It is shown below that for any quaternion and any three dimensional vector in quaternion form, the quaternion product of the quaternion and the vector will result in a vector which is orthogonal to the original quaternion. Furthermore, given any pair of quaternions it is possible to express the one as the product of a unique vector and the other quaternion.

a. Orthogonal Quaternion Theorem

Let p and q be any two quaternions. Then p is orthogonal to q if and only if p is the quaternion product of q and a unique vector v (real part equal to zero) where v is given by

$$v = q^{-1}p \qquad (5.48)$$

Proof:

Let q be any quaternion given by

$$q = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} \qquad (5.49)$$

and let v be any vector in quaternion form which is given by

$$v = \begin{bmatrix} 0 \\ x \\ y \\ z \end{bmatrix} \qquad (5.50)$$

Taking the quaternion product of q and v produces

$$\begin{aligned} qv &= (-q_1x - q_2y - q_3z) \\ &\quad + i(q_0x - q_3y + q_2z) \\ &\quad + j(q_3x + q_0y - q_1z) \\ &\quad + k(-q_2x + q_1y + q_0z) \\ &= \begin{bmatrix} -q_1x - q_2y - q_3z \\ q_0x - q_3y + q_2z \\ q_3x + q_0y - q_1z \\ -q_2x + q_1y + q_0z \end{bmatrix} \end{aligned} \tag{5.51}$$

The dot product of q and the result from Eq. (5.51) is

$$\begin{aligned} (qv) \bullet q &= q \bullet (qv) \\ &= \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} \bullet \begin{bmatrix} -q_1x - q_2y - q_3z \\ q_0x - q_3y + q_2z \\ q_3x + q_0y - q_1z \\ -q_2x + q_1y + q_0z \end{bmatrix} \\ &= q_0(-q_1x - q_2y - q_3z) + q_1(q_0x - q_3y + q_2z) \\ &\quad + q_2(q_3x + q_0y - q_1z) + q_3(-q_2x + q_1y + q_0z) \\ &= -q_0q_1x - q_0q_2y - q_0q_3z + q_1q_0x - q_1q_3y + q_1q_2z \\ &\quad + q_2q_3x + q_2q_0y - q_2q_1z - q_3q_2x + q_3q_1y + q_3q_0z = 0 \end{aligned} \tag{5.52}$$

Thus proving that q and qv are orthogonal for any v.

Now, suppose p and q are quaternions such that

$$p = qv \tag{5.53}$$

for some vector v. Then multiplying both sides of Eq. (5.53) by the inverse of q will produce

$$q^{-1}p = v \tag{5.54}$$

Substitution of the v given by Eq. (5.54) into Eq. (5.53) results in

$$p = qv = q(q^{-1}p) = (qq^{-1})p = p \tag{5.55}$$

Thus given any pair of orthogonal quaternions, one can be written as the quaternion product of the other and a unique vector.

Q.E.D.

According to the above theorem it follows that Δq can be written in the form

$$\Delta q = qv \tag{5.56}$$

where v is a vector in quaternion form such as

$$v = \begin{bmatrix} 0 \\ v_1 \\ v_2 \\ v_3 \end{bmatrix} \tag{5.57}$$

and Δq will be orthogonal to q. Using a Taylor series approximation, the computed measurement vector given by Eq. (5.8) can be approximated for orthogonal Δq by

$$\vec{y}(q+\Delta q) \cong \vec{y}(q) + \frac{\partial y}{\partial q}\Delta q = \vec{y}(q) + \frac{\partial y}{\partial q}(qv) \tag{5.58}$$

Consequently, as v changes, using the chain rule for partial derivatives,

$$\frac{\partial \vec{y}}{\partial v_1} = \frac{\partial \vec{y}}{\partial \vec{q}}\frac{\partial \vec{q}}{\partial v_1} = X\left(q\begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}\right) \tag{5.59}$$

where X is the gradient of $\vec{y}$ with respect q and is derived in Appendix B. Similarly

$$\frac{\partial \vec{y}}{\partial v_2} = \frac{\partial \vec{y}}{\partial \vec{q}}\frac{\partial \vec{q}}{\partial v_2} = X\left(q\begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}\right) \tag{5.60}$$

and

$$\frac{\partial \vec{y}}{\partial v_3} = \frac{\partial \vec{y}}{\partial \vec{q}}\frac{\partial \vec{q}}{\partial v_3} = X\left(q\begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}\right) \tag{5.61}$$

Equations (5.59), (5.60), and (5.61) can be used to define a new 6 x 3 gradient matrix in which each of the equations forms a column of the matrix

$$X_v = \left[\frac{\partial \vec{y}}{\partial v_1}\middle|\frac{\partial \vec{y}}{\partial v_2}\middle|\frac{\partial \vec{y}}{\partial v_3}\right] \tag{5.62}$$

This matrix linearizes $\vec{y}(q+\Delta q)$ with respect to orthogonal Δq and can therefore be used to compute an optimal Δv as

$$\Delta v = \left[X_v{}^T X_v\right]^{-1} X_v{}^T \varepsilon(\hat{q}) \tag{5.63}$$

from which it follows that the optimal Δq under the linearity assumptions is

$$\Delta q = q(0, \Delta v) \tag{5.64}$$

Evaluation of Eq. (5.63) requires inversion of a 3 x 3 matrix rather than inversion of the 4 x 4 X matrix used in Eq. (5.11). Note that normalization of $\hat{q}$ to unit quaternion form will still be required to correct the $O(\Delta q^2)$ effects in Eq. (5.47).

4. Differential Weighting of Sensor Data

Due to noise and interference from electromagnetic sources, magnetometer data is not as reliable as that produced by accelerometers being used to sense gravity. Differential weighting of sensor data allows less weight or confidence to be placed in the magnetometer data relative to that of the accelerometers. This approach makes sense since small drift errors in the horizontal plane are acceptable in most human body tracking applications as long as they are gradual and transient. The effects of noise on the data from a sensor can be expressed using a weighting factor. This factor can be used to implement a weighted least-squares regression analysis algorithm.

If it is assumed that each input parameter is affected by an uncorrellated noise source, the *weighted* modeling error can be written

$$\vec{\varepsilon}(\hat{q}) = \left[\frac{\hat{h}_1 - h_1}{w_1}\ \frac{\hat{h}_2 - h_2}{w_2}\ \frac{\hat{h}_3 - h_3}{w_3}\ \frac{\hat{b}_1 - b_1}{w_4}\ \frac{\hat{b}_2 - b_2}{w_5}\ \frac{\hat{b}_3 - b_3}{w_6}\right]^T \tag{5.65}$$

where w_i is a weighting factor. If it is further assumed that the noise magnitude does not differ for sensors of the same type, the weighted modeling error may be rewritten as

$$\vec{\varepsilon}(\hat{q}) = \left[\hat{h}_1 - h_1\ \hat{h}_2 - h_2\ \hat{h}_3 - h_3\ \frac{w_h}{w_b}(\hat{b}_1 - b_1)\ \frac{w_h}{w_b}(\hat{b}_2 - b_2)\ \frac{w_h}{w_b}(\hat{b}_3 - b_3)\right]^T \tag{5.66}$$

$$= \left[\hat{h}_1 - h_1\ \hat{h}_2 - h_2\ \hat{h}_3 - h_3\ \rho(\hat{b}_1 - b_1)\ \rho(\hat{b}_2 - b_2)\ \rho(\hat{b}_3 - b_3)\right]^T \tag{5.67}$$

where w_h is a weighting factor for accelerometer data and w_b is a weighting factor for the magnetometers. The ratio ρ controls the relative weight placed on the accelerometer and magnetometer data and it will generally be between zero and one. Increasing ρ above unity will cause more weight to placed on the magnetometer data. Decreasing it below one indicates that there is more confidence in accelerometer data. In this case the weighted criterion function becomes

$$\varphi = (\hat{h}_1 - h_1)^2 + (\hat{h}_2 - h_2)^2 + (\hat{h}_3 - h_3)^2 + \rho(\hat{b}_1 - b_1)^2 + \rho(\hat{b}_2 - b_2)^2 + \rho(\hat{b}_3 - b_3)^2 \qquad (5.68)$$

$$= ((\hat{h}_1^{\ 2} - 2\hat{h}_1 h_1 + h_1^{\ 2}) + \ldots + (\rho^2 \hat{b}_3^{\ 2} - 2\rho^2 \hat{b}_3 b_3 + \rho^2 b_3^{\ 2})) \qquad (5.69)$$

Using Eq. (5.69) to derive the error criterion function, results in a modified X matrix given by

$$X = \begin{bmatrix} \frac{\partial \hat{h}_1}{\partial \hat{q}_0} & \frac{\partial \hat{h}_1}{\partial \hat{q}_1} & \frac{\partial \hat{h}_1}{\partial \hat{q}_2} & \frac{\partial \hat{h}_1}{\partial \hat{q}_3} \\ \frac{\partial \hat{h}_2}{\partial \hat{q}_0} & \frac{\partial \hat{h}_2}{\partial \hat{q}_1} & \frac{\partial \hat{h}_2}{\partial \hat{q}_2} & \frac{\partial \hat{h}_2}{\partial \hat{q}_3} \\ \frac{\partial \hat{h}_3}{\partial \hat{q}_0} & \frac{\partial \hat{h}_3}{\partial \hat{q}_1} & \frac{\partial \hat{h}_3}{\partial \hat{q}_2} & \frac{\partial \hat{h}_3}{\partial \hat{q}_3} \\ \rho^2\frac{\partial \hat{b}_1}{\partial \hat{q}_0} & \rho^2\frac{\partial \hat{b}_1}{\partial \hat{q}_1} & \rho^2\frac{\partial \hat{b}_1}{\partial \hat{q}_2} & \rho^2\frac{\partial \hat{b}_1}{\partial \hat{q}_3} \\ \rho^2\frac{\partial \hat{b}_2}{\partial \hat{q}_0} & \rho^2\frac{\partial \hat{b}_2}{\partial \hat{q}_1} & \rho^2\frac{\partial \hat{b}_2}{\partial \hat{q}_2} & \rho^2\frac{\partial \hat{b}_2}{\partial \hat{q}_3} \\ \rho^2\frac{\partial \hat{b}_3}{\partial \hat{q}_0} & \rho^2\frac{\partial \hat{b}_3}{\partial \hat{q}_1} & \rho^2\frac{\partial \hat{b}_3}{\partial \hat{q}_2} & \rho^2\frac{\partial \hat{b}_3}{\partial \hat{q}_3} \end{bmatrix} \qquad (5.70)$$

5. Reduced Rate Drift Correction

The upper loop of Figure 10 serves to correct rate sensor drift and is essentially a low-pass filter. While an attitude update using rate sensor inputs only requires a quaternion multiplication and a single integration, calculating a drift correction requires a matrix inversion and numerous scalar multiplications. If the drift time constant of the rate sensors is long enough and the noise level is low, a drift correction may not be required on every filter cycle.

Eliminating the need to perform drift calculation on every filter cycle leads to a significant reduction in computational costs of running the filter. This reduction may be taken advantage of in two different ways. Reducing the number of drift corrections can be used to increase the overall update rate of a filter. This may result in a reduction in lag and increase in the overall accuracy and resolution of the system. Increasing the drift correction interval can also be used to reduce the number of calculations associated with an individual filter. In a system in which a single processor is being used to run multiple filters, this reduction effectively increases the number of filters which may be operated. For instance on a system which is only capable of running three filters at 100 Hz and performing a drift correction on every filter cycle, it may be possible to run a much larger number of filters by sequencing the drift corrections so that they are only performed for a subset of three of the filters on any given update of posture. If the filter time constant is one second, it may be possible to operate 100 filters at 100 Hz simultaneously with each filter only performing a drift correction after every 100 update cycles.

C. FILTER SIMULATION

Linear analysis provides a method of estimating the response of the filter if the initial orientation estimate, $\hat{q}$, is inaccurate. Such analysis implies that any transient errors in $\hat{q}$ resulting from erroneous initialization will persist for a time inversely proportional to the k used in Eq. (5.14). Specifically, the time constant τ is given by Eq. (5.29) as

$$\tau_{\Delta q} = \frac{1}{k}$$

Let δ_x be a small quantity representing an initial error in the x component of $\hat{q}$. From Eq. (5.30) the resulting errors in the x component of $\hat{q}$ over time will be given by

$$\varepsilon_{\hat{x}}(t) = \delta_x e^{-\tau/\tau_{\Delta q}}$$

Thus, it can be predicted that any error will be reduced to 37% of the initial value by the time $t = \tau_{\Delta q}$. Similar results apply to δ_y and δ_z.

Figure 15 is an example plot of simulation results obtained from an earlier version of the filter [Ref. 6.]. Since these nonlinear simulation results are in close agreement with linear theory, the validity and value of linearization is established. White noise simulation shows noise reduces accuracy, but the filter still works well.

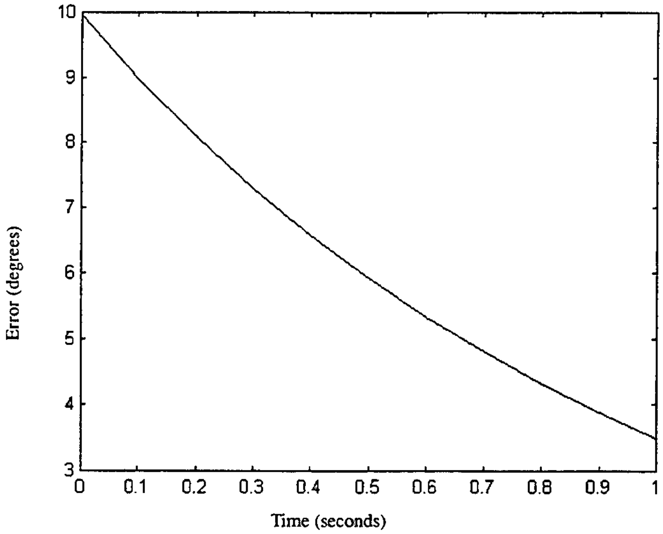


Figure 15: Simulated Nonlinear Filter Response,10 Degree Offset, α=0.1, Δt=0.1 From [Ref. 6.]

Simulation tests performed using noiseless synthetic data and a random starting point were reported in [Ref. 51.] In these trials no failures to converge were observed after ten cycles of Gauss-Newton iteration. Further simulations were conducted in [Ref. 51.] to examine the convergence properties of the filter. In these experiments, the rms (root mean square) accuracy of Gauss-Newton iteration was evaluated as a function of max-noise and the required number of cycles of iteration to achieve convergence. These results confirmed that even with noise levels exceeding 10%, the length of the vector error in q remained at only approximately 80% of the maximum data component noise level.

D. SUMMARY

This chapter describes a quaternion based complementary attitude filter. The filter is designed to accept sensor data from a nine-axis MARG sensor and produce a quaternion representation of the orientation of a tracked rigid body. Due to the use of quaternions, the algorithm described is inherently free from orientation singularities. Continuous correction of drift regardless of the type of motion being tracked is achieved using Gauss-Newton iteration. This property of the filter makes it particularly applicable to human body tracking applications which commonly include short cyclic periods of high linear acceleration.

The algorithm relies upon the Orthogonal Quaternion Theorem. The theorem both resolves the singularity problem of Gauss-Newton iteration applied to quaternion orientation tracking and reduces the size of the associated regression matrix from 4 x 4 to 3 x 3. This reduction results in a significant computational advantage since the inversion of the regression matrix is probably the most time consuming part of the drift correction process. This improvement is especially important when simultaneously tracking a large number of human limb segments or when implementing the algorithm on imbedded microprocessors.

The described algorithm also includes two scalar gain factors that allow "tuning" of the filter to fit a particular tracking situation. Guidelines for choosing values for these parameters are provided, but it is believed that final selection of gains is best accomplished by adjustment during the course of an experiment. It is conjectured that periods between drift corrections can be extended resulting in either a higher update rate or the ability to implement a greater number of filters simultaneously using less computing power.

The quaternion attitude filter fulfills the need for an efficient and robust algorithm for sourceless real-time tracking of human limb segments without the computational complexity of previous Euler angle based algorithms designed for head tracking or ship and aircraft navigation systems.

VI. IMPLEMENTATION OF INERTIAL AND MAGNETIC TRACKING OF HUMAN LIMB SEGMENTS

A. INTRODUCTION

This chapter describes pertinent details of an implementation of a prototype system for tracking human body motions using magnetic, angular rate, and gravity sensors. The central data processing algorithm is the quaternion attitude filter described in the previous chapter. The goal of the system is to demonstrate the practicality and robustness of inertial and magnetic orientation tracking as well as to provide a test-bed for further experiments and future system development. Several features are considered imperative if these goals are to be meet. Among these are

- Orientation tracking of any three or more human limb segments using nine-axis MARG sensors
- Sufficient dynamic response and update rate (100 HZ or better) to capture faster human body motions
- Ability to change quaternion filter operating parameters while the system is in operation
- Calibration of individual sensors without the use of any specialized equipment
- Simplified human kinematic model based entirely on quaternions capable of accepting orientation parameters relative to an earth fixed reference frame in quaternion form
- Automatic accounting for the peculiarities related to the mounting of a sensor on an associated limb segment
- Adjustable human model to take into account anthropometric variations between different individuals
- Creation of data files for recording data relating to posture estimation as well as filter operation
- Archiving of system configurations for retrieval and further experimentation

Figure 16 is a diagram of the prototype system. Depicted are three body-mounted MARG sensors outputting analog signals to three I/O connection boards. The output from each connection board is digitized by an associated A/D converter card. The cards themselves are mounted in a standard Wintel desktop computer. All data processing and rendering calculations are performed by software running on this single processor machine. The display monitor provides a means of visually displaying the estimated posture of the tracked individual. The principal components of the system are discussed in detail in the following sections.

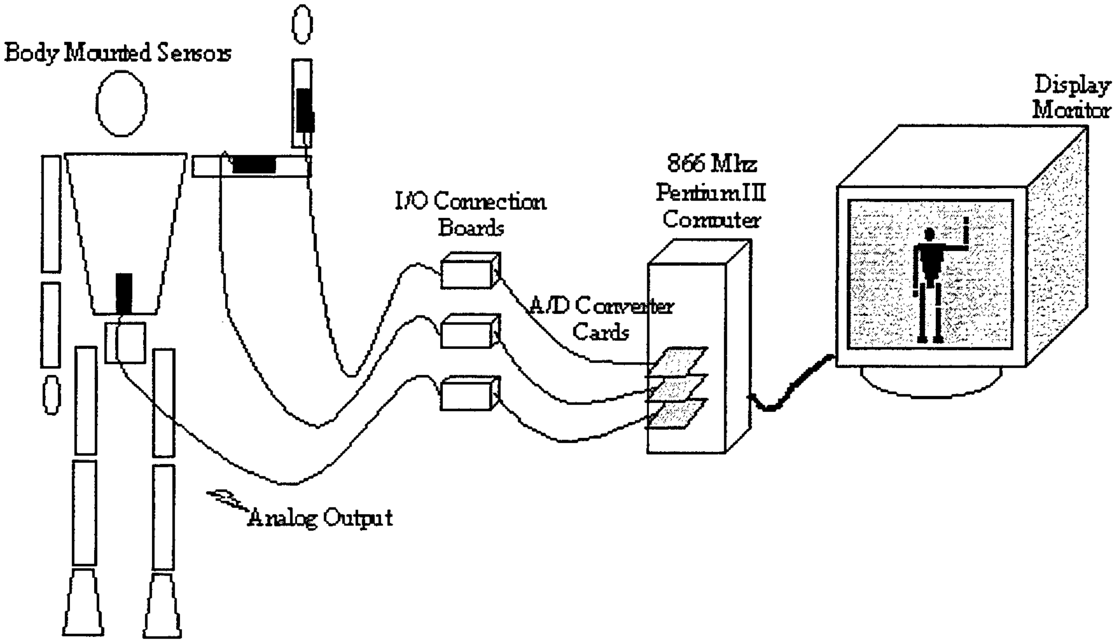


Figure 16: Prototype Inertial and Magnetic Body Tracking System

B. PROTOTYPE MARG SENSORS

The prototype MARG sensors used in this research were custom built using off-the-shelf, low cost components. No significant attempt was made to produce an extremely small sensor. Ease of use and construction were the overriding factors affecting sensor design. These sensor components are housed in a lightweight case constructed of birch wood to prevent shock damage and to provide a stable temperature environment for the rate sensors (Figure 17). The case material was also chosen to take advantage of its lack of magnetic properties.

The MARG sensors units are designed to combine three mutually orthogonal magnetometers, three mutually orthogonal angular rate sensors, and three mutually orthogonal accelerometers into a single compact package. To track the entire human body, approximately fifteen of these nine-axis units would be required. One sensor would be attached to each limb segment to be tracked. The exact number of sensors needed would depend upon the desired motion tracking detail to be captured. Three such sensors were used in the system described in this research.

Each sensor package measures 10.1 x 5.5 x 2.5 cm. The analog output of the sensor is connected to a breakout header via a thin VGA monitor cable. Output range is 0-5 vdc. The power requirement of the sensors is 12 vdc at approximately 50 milliamperes. The primary sensing components are a Crossbow CXL04M3 triaxial accelerometer [Ref. 18.], a Honeywell HMC2003 3-axis magnetometer [Ref. 39.] and three Tokin CG-16D series miniature angular rate sensors mounted in an orthogonal configuration [Ref. 84.]. The individual components are integrated using a single integrated circuit board with the accelerometers mounted separately. The circuit provides a set/reset circuit capability for the magnetometers and allows manual adjustment of magnetometer null points. Rate sensor output voltage is amplified by a factor of five to attenuate rate sensor oscillator noise. All three sensors were fabricated by McKinney Technology of Prunedale, California [Ref. 61.].

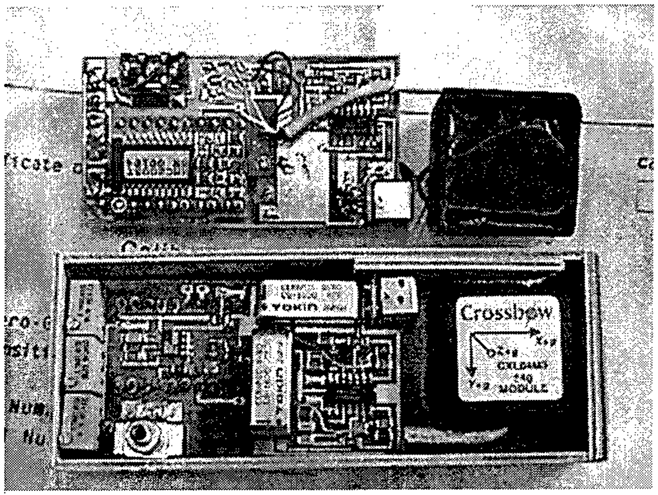


Figure 17: Prototype MARG Sensor From [Ref. 61.]

1. Sensor Components

a. Crossbow CXL04M3 Triaxial Accelerometer

The CXL04M3 triaxial accelerometer package contains three silicon micromachined Analog Devices ADXL05 accelerometers [Ref. 3.] mounted in an orthogonal configuration. The ADXL05 is a force balanced capacitive accelerometer with capability to measure dc accelerations which are typical of an inertial force such as gravity. When oriented to the earth's gravity, *x* axis pointing up, the accelerometer will experience a positive 1g acceleration. Full scale output is selectable from +/- 1 to +/- 5g. Dimensions of the triaxial package are approximately 25 x 25 x 19mm. Individual accelerometer cans have a diameter of 9.4mm and a height of 4.7 mm. Shock survival is 1000g when unpowered, 500g powered. Additional pertinent characteristics of the CXL04M3 are given in Table 1.

Characteristic	Range	Units
Zero g Output	2.5 +/- 0.1	Volts
Output Voltage	0 - 5	Volts
Sensitivity	500 +/- 5%	mV/g
Noise	5	mg rms
Bandwidth	DC-100	Hz
Temperature Range	-40 to +85	C
Supply Voltage	5 +/- 0.25	VDC
Supply Current	24	mA

Table 1: CXL04M3 Triaxial Accelerometer Specifications After [Ref. 18.]

b. Tokin CG-16D Series Rate Gyros

The Tokin CG-16D is a ceramic angular rate sensor composed of a single piezoelectric ceramic column printed with electrodes [Ref. 84.]. It is primarily designed for use as a vibratory gyroscope in vehicle navigation systems. The advertised maximum detectable angular rate is given as +/- 90 degrees per sec. Though the response may no longer be linear, higher rates have been observed in experiments. Sensor dimensions are given as 8 x 8 x 20 mm. Shock survival is 300g. Three CG-16D angular rate sensors are mounted in an orthogonal configuration inside each MARG sensor. Due to the unstable characteristics of the sensors under temperature changes, internal MARG rate sensor circuitry amplifies the sensor output and performs temperature compensation to maintain null output voltage at a constant value. Additional pertinent characteristics of the CG-16D are given in Table 2.

Characteristic	Range	Units
Reference Voltage	2.4	Volts
Output Voltage	0 - 5	Volts
Sensitivity	1.1 +/- 20%	mV/deg./sec.
Output Voltage at zero angular rate (25 degrees C)	+/ 300	mVolts
Output Voltage at zero angular rate (any Temp.)	+/-500	mVolts
Bandwidth	100	Hz
Temperature Range	-5 to +76	C
Supply Voltage	5	VDC
Supply Current	7	mA

Table 2: CG-16D Ceramic Rate Gyro Specifications After [Ref. 84.]

c. Honeywell HMC2003 3-Axis Magnetometer

The Honeywell HMC2003 is a solid state 3-axis magnetometer contained in a 20-pin hybrid DIP package [Ref. 39.]. The local magnetic field is measured by three permalloy magnetoresistive (MR) Honeywell HMC1001/2 microcircuits which convert magnetic fields to a differential output voltage. The transducer is configured as a magnetoresistive Wheatstone bridge. Two "straps", OFFSET and Set/Reset, eliminate the need for external coils. DIP footprint is approximately 25 x 19 mm. Shock survival is 100g. Pertinent characteristics of the CG-HMC2003 are given in Table 3.

Characteristic	Range	Units
Field Range	-2 to 2	gauss
Output Voltage	0.5 - 4.5	Volts
Null Field Output	2.5	Volts
Sensitivity	1	V/gauss
Bandwidth	1000	Hz
Temperature Range	-40 to +85	C
Supply Voltage	6 - 15	VDC
Supply Current	20	mA

Table 3: Honeywell HMC2003 Three-Axis Magnetic Sensor Hybrid Specifications After [Ref. 39.]

2. Magnetometer Set/Reset

Early system testing was hampered due to saturation of the MARG sensor magnetometers by small magnetic fields. Saturation produced flips or reversals resulting in changes in the sensor characteristics. Once saturated, the lack of a built-in reset made it difficult to restore the magnetic sensors to a usable condition. Only through repeated exposure to various magnetic fields and trail and error iterations could the sensors be returned to a functional condition. Often, the magnetometer null points had changed following these procedures making it necessary to recalibrate the sensor.

Manufacturers literature states that HMC1001/2 magnetometer saturation occurs due to the influence of a strong magnetic field in excess of 30 gauss which can cause the polarity of the MR film magnetization to flip [Ref. 15.]. In practice, changes in the magnetometer characteristics were found to occur in the presence of weaker fields such as those caused by exposure to metal scissors or cell-phones. Following such an upset field, a strong restoring magnetic field must be momentarily applied to restore, or set, the sensor characteristics. The effect is commonly referred to as applying a *set* or *reset* pulse. The Honeywell HMC1001/2 incorporates a patented on-chip strap for performing the re-magnetization electrically. This flipping may be performed manually or automatically at various time intervals.[Ref. 15.]

The prototype MARG sensors used in this research incorporate a manual set/reset circuit to electrically restore the magnetometers to proper operation. Activation of the circuit is accomplished using a sensor mounted button. The associated circuit is depicted in Figure 18. The purpose of the circuit is to set or reset the permalloy film contained in the individual magnetometers by applying a current pulse of 3-4 amps for approximately 20-50 nsec.

3. Analog to Digital Conversion

Analog sensor output signals must be converted to digital form in order to perform processing using a digital computer. In this research, analog to digital conversion of sensor output voltages was completed external to the sensors using one National Instruments PCI-MIO-16XE-50 data acquisition card for each MARG sensor. Each data acquisition card was inserted into a PCI slot on the mother board of the data processing computer. The PCI-MIO-16XE-50 is a 16-bit A/D converter capable of sampling either 16 single-ended or 8 double-ended analog input channels. Maximum sampling rate is 20K samples/sec. Input voltage ranges are 0 - 10V in single ended mode and -10 to 10V in double sided mode. The boards are completely Plug and Play, multifunction analog, digital, and timing I/O boards for PCI bus computers. [Ref. 69.] Sensor to board connection was completed using a National Instruments SCB68 type I/O connection board.[Ref. 68.]

4. Data Processing

The prototype inertial and magnetic body tracking system depicted in Figure 16 uses an Intel based desktop computer to complete all data processing and rendering

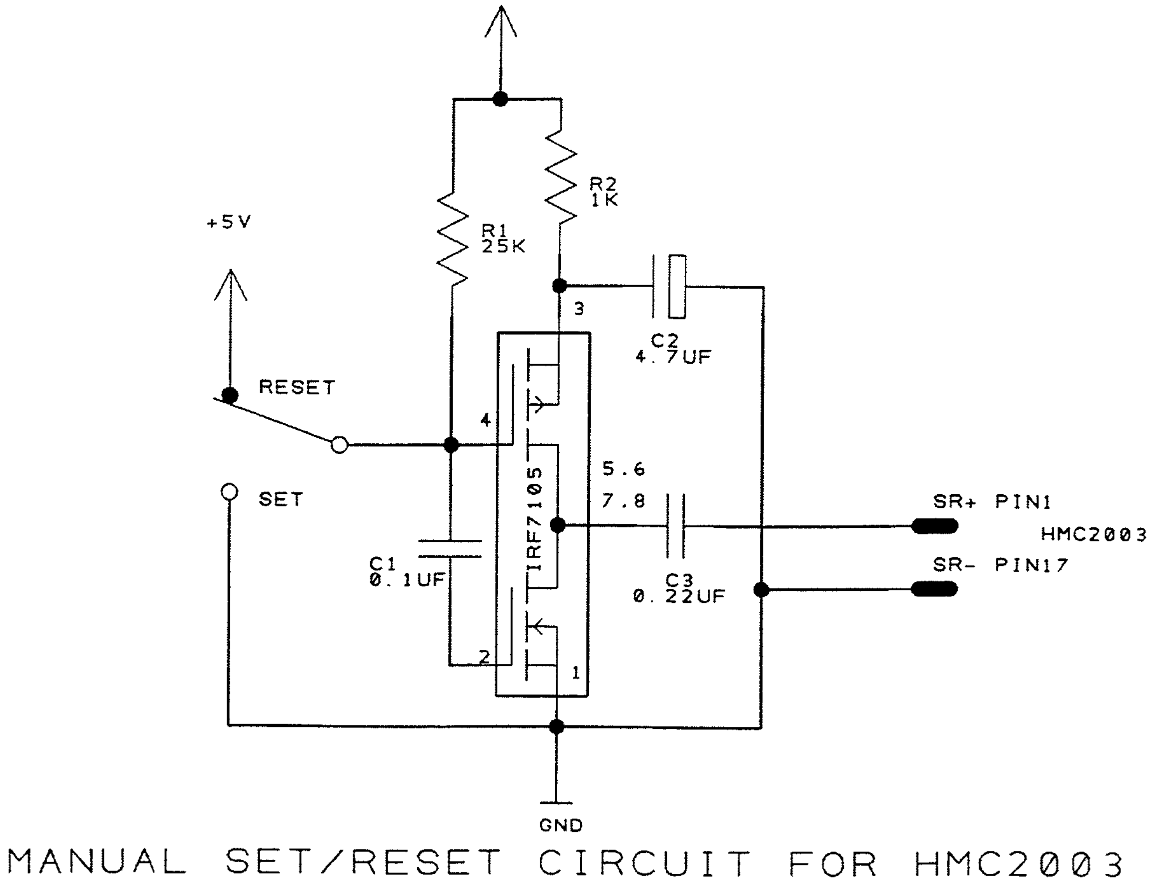


Figure 18: MARG Sensor Magnetometer Set/Reset Circuit Schematic From [Ref.

functions. The computer was manufactured by Micron Electronics, Inc. The single installed CPU is an 866 MHz Intel Pentium III. The machine contains 256 MB of RAM. Hardware rendering is performed by a NVIDIA GeForce2 GTS video card. The Microsoft Windows 2000 operating system is used in order to achieve accurate timing of body tracking system events.

C. SYSTEM SOFTWARE

The system software implements the estimation as well as calibration algorithms which make possible tracking of human body segments using MARG sensors. Drift correction is performed using the reduced order form of Gauss-Newton iteration described in the previous chapter. Facilities are included to allow performance of experiments related to differential weighting of magnetometer and accelerometer data, variation of intervals between drift corrections, and adjustment of the filter gains. A sensor calibration algorithm allows system users to calibrate individual sensors by subjecting them to a series of six 90 or 180 degree rotations followed by two 360 degree rotations oriented with respect to the local magnetic field.

In addition, the system software includes a fully articulated human model based entirely on quaternion/vector pairs. No rotation matrices are used to position the model. Limb segments are oriented independently of one another and positioned through the addition of limb associated vectors. Limb segment lengths are fully adjustable to allow compensation for variation in the relative dimensions of limb segments for different individuals. The model is positioned and oriented relative to a *z* axis down coordinate system [Ref. 52.].

The system software is fully serialized allowing for archival of experimental configurations with varying model dimensions and differing filter parameter settings. Facilities are provided for creating files containing data related to full body posture estimation or data related to the operation of an individual filter object.

The body tracking software for this research was designed using object oriented techniques. All code was written using the Microsoft Visual C++ Integrated Development Environment (IDE) and compiled under the Visual C++ 6.0 compiler. The application is a Single Document Interface (SDI) which follows the Microsoft Foundation Class (MFC) Document/View architecture and application framework conventions. The code is single threaded. Estimation and rendering events are window system timer driven at 100 Hz and 25 Hz respectively.

Figure 19 is a simplified class diagram of the body tracking software. Minor dialog box classes and other user interface classes have been omitted. For clarity, class methods and data members are not individually listed. In viewing the figure, the classes can be separated into two groups, those under the application document class, *CBodyTrackingDoc*, and those under the application view class, *CBodyTrackingView*.

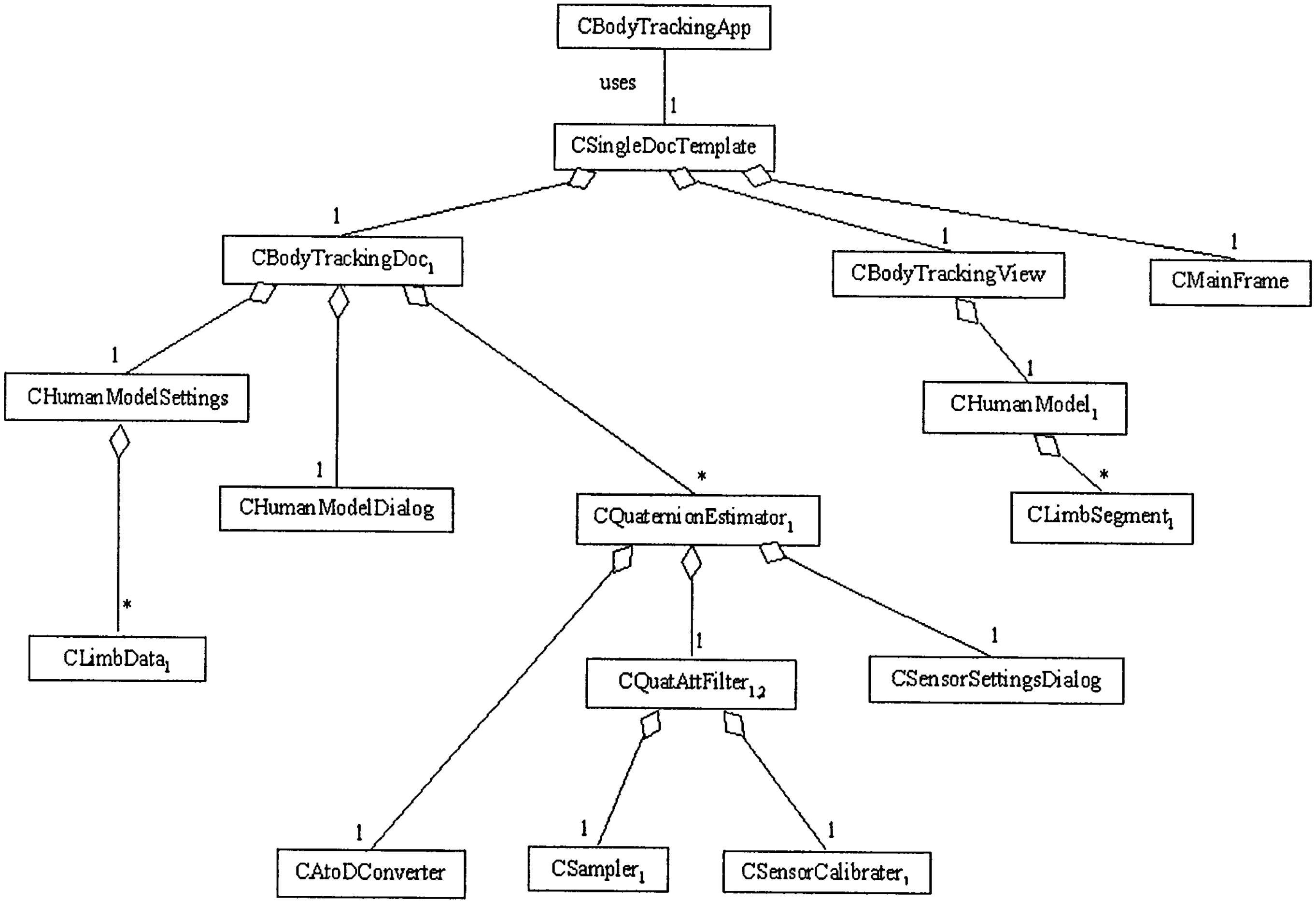


Figure 19: Body Tracking Software Simplified Class Diagram

1. uses Quaternion Class
2. uses Matrix Class

The classes under the application document class, CBodyTrackingDoc, are related to the generation and saving of data as well as the system configuration. These classes include *CAtoDConverter* which retrieves sensor data from the system hardware, *CSampler* which formats data for submission to the quaternion filter, *CQuatAttFilter* which implements the quaternion based attitude filter algorithm and *CHumanModelSettings* which holds data related to the posture and configuration of the human model. *CSensorCalibrater* implements the MARG sensor calibration algorithm. *CQuaternionEstimator* serves as a container class to facilitate object communication. *CLimbData* objects are used to hold the current length and orientation data of individual limb segments. *CHumanModelDialog* and *CSensorSettingDialog* objects allow user adjustment of application settings. All document related classes are serialized.

The classes under the application view class, CBodyTrackingView are responsible for providing a view of the data of the application. These data are contained in the document. In the case of the body tracking system, all data pertains to the orientation, location, and size of human model limb segments. The *CHumanModel* class implements a human model using objects of type *CLimbSegment.* The number of CLimbSegment objects used is determined by the number of links in the model.

Figure 20 depicts the major data flow paths between the instantiated objects of the system. The primary input to the system is nine-axis MARG sensor data. The state of the system may also be affected by the user through the use of dialog boxes. System outputs are not depicted. These include visual display of the posture of the articulated human model and the creation of data files for post-processing or plotting. MARG sensor data is only received by a CSensorCalibrater object when the associated sensor is being calibrated.

The following sections describe the key classes and algorithms implemented in more detail.

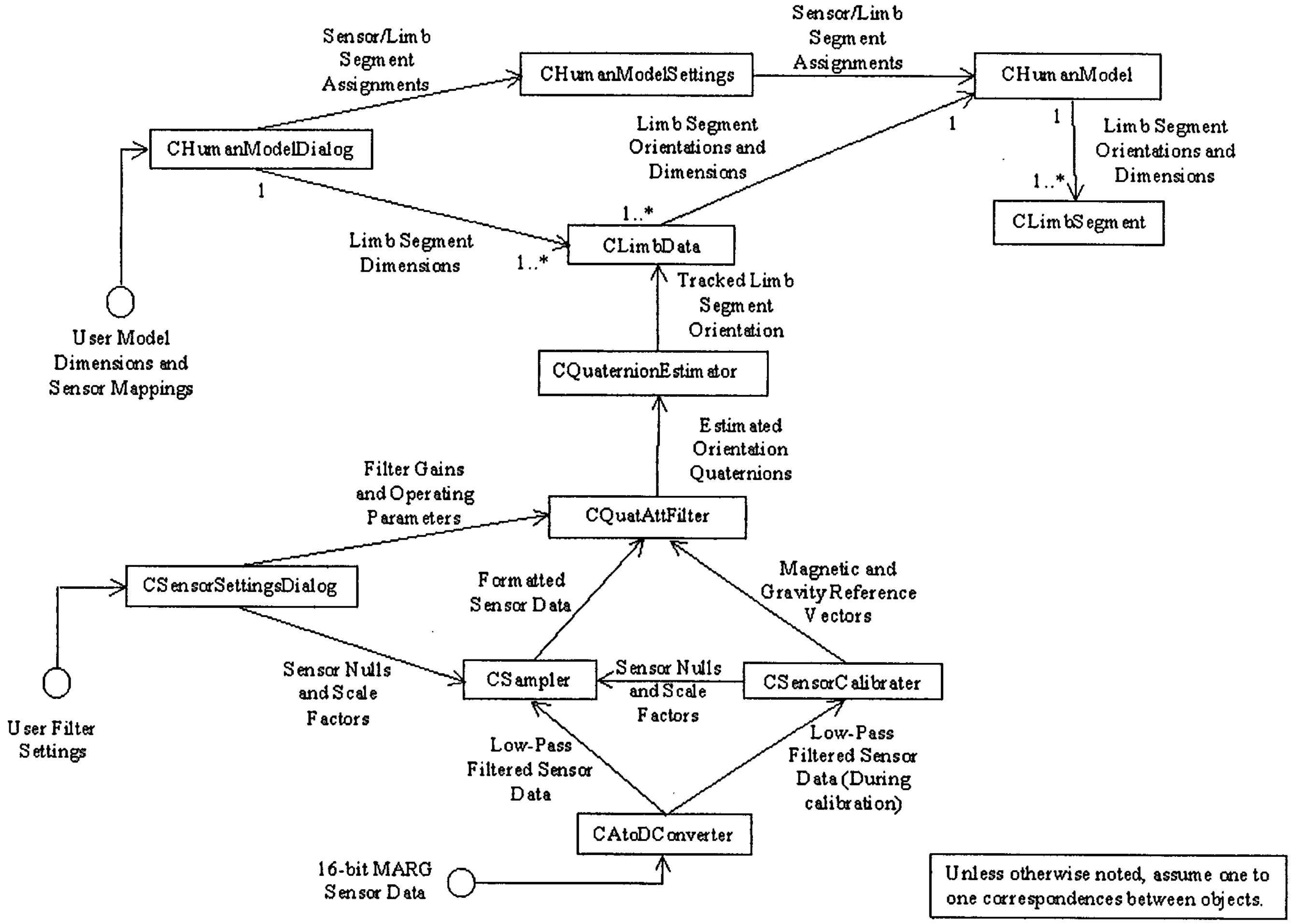

Sensor/Limb Segment Assignments
CHumanModelSettings
Sensor/Limb Segment Assignments
CHumanModel
CHumanModelDialog
Limb Segment Orientations and Dimensions
1
1
Limb Segment Orientations and Dimensions
1..*
1
1..*
CLimbSegment
Limb Segment Dimensions
1..*
CLimbData
Tracked Limb Segment Orientation
User Model Dimensions and Sensor Mappings
CQuaternionEstimator
Estimated Orientation Quaternions
Filter Gains and Operating Parameters
CQuatAttFilter
CSensorSettingsDialog
Formatted Sensor Data
Magnetic and Gravity Reference Vectors
Sensor Nulls and Scale Factors
CSampler
Sensor Nulls and Scale Factors
CSensorCalibrater
User Filter Settings
Low-Pass Filtered Sensor Data
Low-Pass Filtered Sensor Data (During calibration)
CAtoDConverter
16-bit MARG Sensor Data
Unless otherwise noted, assume one to one correspondences between objects.

Figure 20: Class Instance Data Flow Diagram

1. Quaternion Filter

The CQuatAttFilter class implements the reduced order quaternion attitude filter described in Chapter V using the simplified *X* matrix derived in Appendix B. The filter is complementary in form. Estimation error is minimized using Gauss-Newton iteration. Options are included for performing differential weighting of sensor data and reduced rate drift correction. Reduced rate drift correction may occur at specified time intervals or may be applied to the system filter objects in a round-robin fashion in order to allow a greater number of filters to operate at higher update rates. Matrix and quaternion mathematical abstractions are handled using objects of the *Matrix* and *Quaternion* classes respectively. With the exception of the measured rate quaternion, *qDot* (Eq. (5.1)), and the correction quaternion *qDotEpsilon* (Eq. (5.14)), all quaternions are normalized to unit length. The reference unit vectors, *m* and *n*, given by Eq. (5.5) and Eq. (5.6) are determined during the calibration process and set by an associated object of the CSensorCalibrater class. Expected input to the class is nine floating point numbers corresponding to the nine analog output voltages of an associated MARG sensor. The angular rate values must be provided in radians per second. Magnetometer and accelerometer readings are used to describe the components of two directional vectors. Only the direction of these vectors is of importance and each is normalized to unit length. Thus, there is no need to follow any particular unit convention

Once the filter object has been instantiated and estimation has begun, the *estimateRotation* method serves as the primary interface to obtain updated orientation estimates. The quaternion returned by this method represents the orientation relative to an Earth-fixed reference frame of the associated MARG sensor block. Figure 21 depicts the control logic flow and the step by step algorithm followed by this method. In viewing the figure, it appears that the computational expense of calculating drift corrections based upon magnetometer and accelerometer data is much higher than merely updating the orientation estimate using only rate sensor data. This is in fact the case. Derivation of the *X* matrix in the "Calculate X Matrix" step requires the computation of multiple partial derivatives (See appendix B) and “Calculate full Delta v step” requires inversion of a 3 x 3 matrix. Filter operating parameters and gains may be adjusted as the filter operates using the dialog shown in Figure 22.

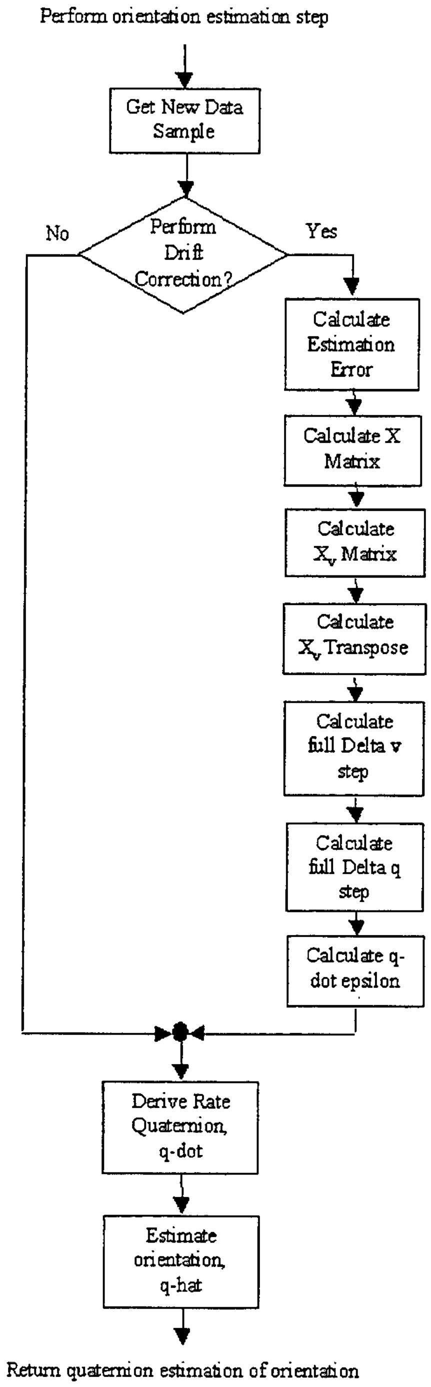
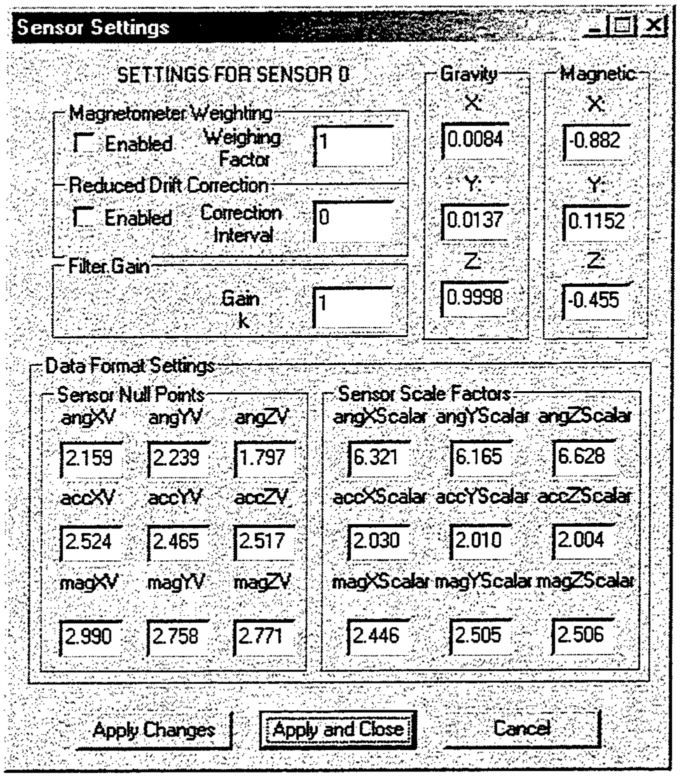

Perform orientation estimation step
Get New Data Sample
Perform Drift Correction?
No
Yes
Calculate Estimation Error
Calculate X Matrix
Calculate X_v Matrix
Calculate X_v Transpose
Calculate full Delta v step
Calculate full Delta q step
Calculate q-dot epsilon
Derive Rate Quaternion, q-dot
Estimate orientation, q-hat
Return quaternion estimation of orientation

Figure 21: Orientation Estimation Flow Chart

2. Sensor Calibration

The accuracy of the orientation estimate produced by the quaternion filter depends heavily on the data which is input. In order for the system to operate properly, it is imperative that the *null point* and *scale factor* of each individual component of the MARG sensors be determined prior to commencing limb tracking. The null point and scale factor for each component are found through a calibration procedure. Practical use of an inertial tracking system requires that this procedure be both efficient and accurate. Unless the characteristics of the sensors themselves change, calibration need only be accomplished once. Magnetometer calibration may need to be accomplished more often due to changes in the local magnetic field. Fortunately, it has been found that slight inaccuracies in the magnetometer readings do not adversely affect the overall operation of the tracking system to the same degree as inaccuracies in accelerometer and rate sensor data.



SETTINGS FOR SENSOR 0

Magnetometer Weighting: □ Enabled, Weighting Factor: 1

Reduced Drift Correction: □ Enabled, Correction Interval: 0

Filter Gain: Gain k: 1

	Gravity	Magnetic
X	0.0084	-0.882
Y	0.0137	0.1152
Z	0.9998	-0.455

Data Format Settings

Sensor Null Points

angXV	angYV	angZV
2.159	2.239	1.797
accXV	accYV	accZV
2.524	2.465	2.517
magXV	magYV	magZV
2.990	2.758	2.771

Sensor Scale Factors

angXScalar	angYScalar	angZScalar
6.321	6.165	6.628
accXScalar	accYScalar	accZScalar
2.030	2.010	2.004
magXScalar	magYScalar	magZScalar
2.446	2.505	2.506

Figure 22: Dialog For Manually Setting Filter Parameters and Sensor Data Null Voltages and

In the body tracking software, the nine digital values corresponding to a given MARG sensor data sample are converted to positive floating point numbers. These numbers are the single-sided voltages which are output by the sensors. Based upon this assumption, each number is formatted for input into the quaternion filter by

$$formatted\ number = (voltage - null\ point) \times scale\ factor \times units \qquad (6.1)$$

In practice the separate *units* term is not necessary since it can be combined with the scale factor and a scalar multiplication can thus be saved.

An individual linear accelerometer can be calibrated by placing it in a vertical position to sense gravity in one direction and then turning it over to sense gravity in the other. Half way between the readings taken is the null point.

$$accel\ \ null = \frac{(accel\ \ max + accel\ \ min)}{2} \tag{6.2}$$

Multiplication of a correct scale factor times the accelerometer output values will result in a product of 1 g in one direction and -1 g in the other. This scale factor can be found using

$$accel\ \ scale = \frac{(accel\ \ units) \times 2}{(accel\ \ max - accel\ \ min)} \tag{6.3}$$

Calibration of a triaxial accelerometer module could be accomplished in a manner similar to that described above. The module would have to be placed in six different positions so that each accelerometer could sense gravity along both its negative and positive axes.

An obvious method of magnetometer calibration is very similar to that used for accelerometers. Instead of orienting each sensor relative to the gravity vector, each magnetometer would have to be placed in a position in which it could sense the maximum strength of the local magnetic field along both its negative and positive axes. This may be accomplished by pointing the magnetometer axis toward the local north and recording the maximum and minimum voltages as the magnetometer is rotated 360 degrees about an axis oriented toward the east. Half way between the maximum and minimum readings obtained is the null point of the magnetometer.

$$mag\ \ null = \frac{(mag\ \ max + mag\ \ min)}{2} \tag{6.4}$$

Multiplication of a correct scale factor times the magnetometer output values should result in a reading of approximately 0.6 gauss in one direction and -0.6 gauss in the other depending upon the actual strength of the local magnetic field.

$$mag\ \ scale = \frac{(mag\ \ units) \times 2}{(mag\ \ max - mag\ \ min)} \tag{6.5}$$

Complete calibration of a three-axis magnetometer could thus be accomplished by performing one such rotation for each individual sensor.

Determination of the null point of an angular rate sensor can be accomplished by recording and averaging over some time period the output of a static sensor. Scale factors are determined by integrating the output of angular rate sensor over time. If an angular rate sensor is subjected to a known angle of rotation and its output is integrated during the period of rotation, the correct scale factor will cause the result of that integration to equal the angle of rotation. The scale factor for a rate sensor can therefore be determined following a known rotation using

$$scale\ factor = \frac{known\ \ rotation}{estimated\ \ rotation} \tag{6.6}$$

where the *estimated rotation* term is the result of integrating the output of the sensor with a scale factor of unity. In practical applications it may be desirable to make several estimates of the scale factor while putting the sensor through several known positive and negative rotations and then averaging the results.

From the above, it is apparent that a MARG sensor could be completely calibrated using a level nonmagnetic platform and a simple compass to indicate the direction of the local magnetic field. The sensor could be calibrated by placing it in the six positions which allow each accelerometer to sense gravitation acceleration in both the positive and negative directions, subjecting each rate sensor to one or more known rotations and rotating the MARG sensor in a manner such that maximum and minimum local magnetic field readings can be obtained for each magnetometer. The following calibration algorithm is implemented in the body tracking software as a state machine. The state machine includes approximately 33 separate states. Rate sensor scale factors are calculated by averaging the estimates produced by one negative and one positive rotation. The steps of the algorithm listed below loosely correspond to the actual physical actions which a person doing the calibration must perform upon the sensor.

Finding Inertial Sensor Null Points and Scale Factors

1. Place the sensor in a stationary position on a flat level nonmagnetic surface with the positive z axis of the sensor pointing down. While the sensor is in this position record the maximum voltage reading for the z-axis accelerometer as *accZMax*. Set the rate sensor null points *angXNull*, *angYNull* and *angZNull* to the rate sensor readings obtained while in this stationary position.
2. Rotate the sensor 90 degrees about the positive x-axis. While performing this rotation integrate the output of the x-axis rate sensor (Figure 23).

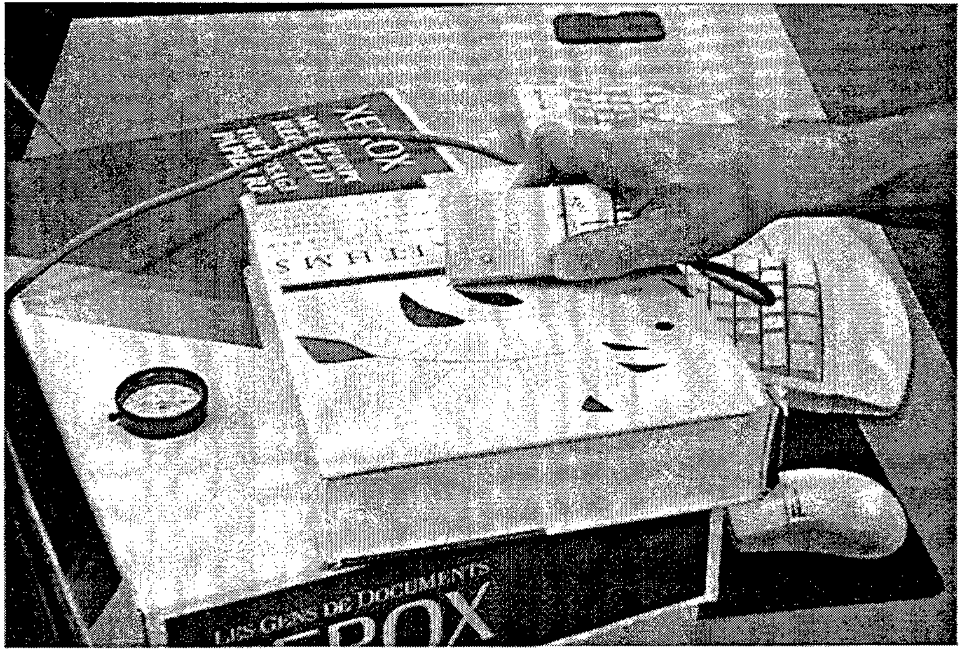


Figure 23: Rotating Sensor 90 Degrees About Positive x-axis For Rate Calibration

3. Following completion of the rotation, record the maximum voltage reading for the y-axis accelerometer as *accYMax*. Make a first estimate of the x-axis rate sensor scale factor, *angScaleXOne*, using Eq. (6.6).
4. Rotate the sensor 180 degrees about the negative x-axis. While performing this rotation integrate the output of the x-axis rate sensor.
5. Following completion of the rotation, record the minimum voltage reading for the y-axis accelerometer as *accYMin*. Make a second estimate of the x-axis rate sensor scale factor, *angScaleXTwo*, using Eq. (6.6). Set the scale factor for the

x-axis rate sensor to the average of *angScaleXOne* and *angScaleXTwo*. Calculate the null point for the *y*-axis accelerometer using Eq. (6.2).

6. Rotate the sensor 90 degrees about the positive *z*-axis. While performing this rotation integrate the output of the *z*-axis rate sensor.
7. Following completion of the rotation, record the minimum voltage reading for the *x*-axis accelerometer as *accXMin*. Make a first estimate of the *z*-axis rate sensor scale factor, *angScaleZOne*, using Eq. (6.6).
8. Rotate the sensor 180 degrees about the negative *z*-axis. While performing this rotation integrate the output of the *z*-axis rate sensor.
9. Following completion of the rotation, record the maximum voltage reading for the *x*-axis accelerometer as *accXMax*. Make a second estimate of the *z*-axis rate sensor scale factor, *angScaleZTwo*, using Eq. (6.6). Set the scale factor for the *z*-axis rate sensor to the average of *angScaleZOne* and *angScaleZTwo*. Calculate the null point for the *x*-axis accelerometer using Eq. (6.2).
10. Rotate the sensor 90 degrees about the negative *y*-axis. While performing this rotation integrate the output of the *y*-axis rate sensor.
11. Following completion of the rotation, record the minimum voltage reading for the *z*-axis accelerometer as *accZMin*. Make a first estimate of the *y*-axis rate sensor scale factor, *angScaleYOne*, using Eq. (6.6). Calculate the null point for the *z*-axis accelerometer using Eq. (6.2).
12. Rotate the sensor 180 degrees about the positive *y*-axis. While performing this rotation integrate the output of the *y*-axis rate sensor.
13. Following completion of the rotation, make a second estimate of the *y*-axis rate sensor scale factor, *angScaleYTwo*, using Eq. (6.6). Set the scale factor for the *y*-axis rate sensor to the average of *angScaleYOne* and *angScaleYTwo*.
14. Calculate the accelerometer scale factors using Eq. (6.3).

Finding Magnetometer Maximum and Minimum Voltage Readings

15. Point the sensor *x*-axis north and rotate the sensor 360 degrees about the *y*-axis. Record the minimum and maximum voltages obtained from the x-axis magne-

tometer during this rotation.

16. Point the sensor y-axis north and rotate the sensor 360 degrees about the x-axis. Record the minimum and maximum voltages obtained from the y-axis and z-axis magnetometers during this rotation.
17. Calculate the magnetometer null points using Eq. (6.4). Calculate the magnetometer scale factors using Eq. (6.5).

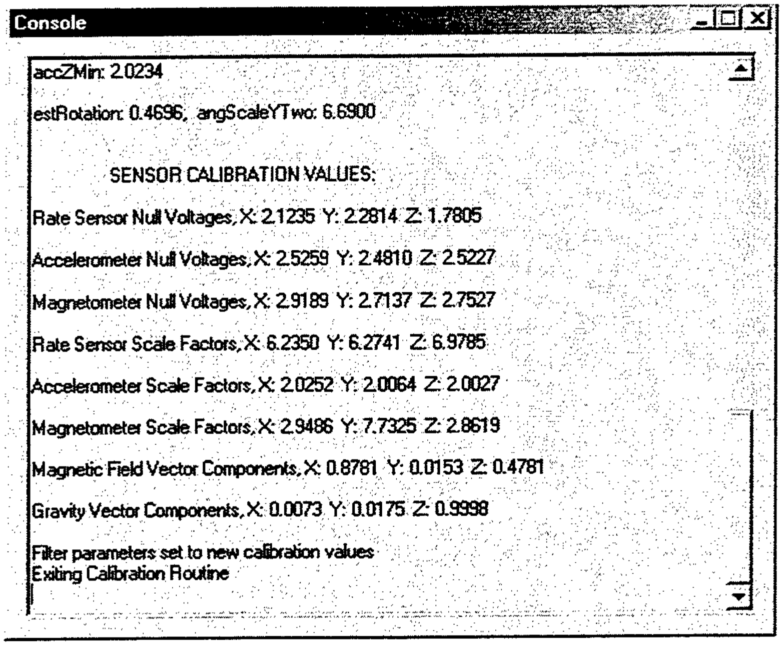
Finding Gravity and Magnetic Reference Vectors

18. Place the sensor in the reference position with the positive x-axis pointing toward magnetic north, positive y-axis east, and the positive z-axis pointing down. While in this stationary position record the reading produced by the magnetometers and accelerometers. Convert these readings using Eq. (6.1). The six numbers produced correspond to the x, y, and z components of the two reference vectors.

Once the sequence of rotations becomes familiar, the entire calibration procedure can be performed in less than one minute. Figure 24 is shows a console display of calibration results.

In the implementation described above each sensor is calibrated individually. The algorithm described could be used to allow calibration of numerous sensors simultaneously. In that case, MARG sensor calibration could be carried out by placing the sensors in a special apparatus before commencing body tracking. The apparatus could be a simple box containing a bin for each sensor. The apparatus could then be put through the same sequence of rotations and orientations as those used for an individual sensor.

Steps 15 through 17 of the calibration procedure could be accomplished separately to prepare the system to operate in a different magnetic environment. It also is possible to change the orientation and magnitude of the rotations performed to allow magnetometer calibration without completing of steps 15 and 16. The maximum and minimum voltage output for each magnetometer could be determined if the rate sensor were calibrated through one positive and one negative 180 degree rotation about the each axis with the axis



```
accZMin: 2.0234

estRotation: 0.4696,  angScaleYTwo: 6.6900


        SENSOR CALIBRATION VALUES:

Rate Sensor Null Voltages,X: 2.1235  Y: 2.2814  Z: 1.7805

Accelerometer Null Voltages,X: 2.5259  Y: 2.4810  Z: 2.5227

Magnetometer Null Voltages,X: 2.9189  Y: 2.7137  Z: 2.7527

Rate Sensor Scale Factors,X: 6.2350  Y: 6.2741  Z: 6.9785

Accelerometer Scale Factors,X: 2.0252  Y: 2.0064  Z: 2.0027

Magnetometer Scale Factors,X: 2.9486  Y: 7.7325  Z: 2.8619

Magnetic Field Vector Components,X: 0.8781  Y: 0.0153  Z: 0.4781

Gravity Vector Components,X: 0.0073  Y: 0.0175  Z: 0.9998

Filter parameters set to new calibration values
Exiting Calibration Routine
```

Figure 24: Console Display Of Sensor Calibration Results

orthogonal to the direction of the local magnetic field. This sequence was not used here due to the position of the sensor data cable and the reset button on the MARG sensor housings.

3. Quaternion Human Body Model

The quaternion human body model is designed to accept orientation data in quaternion form relative to an earth-fixed reference frame. The model posture is set using only vector addition and quaternion rotation. Vector addition determines the position of the inboard end of each limb segment. Quaternion rotation of limb segment vertices is used to set the limb segment attitude. This attitude is set independently of those to which it is attached. No homogeneous transform matrices are used. The model includes no provisions for joint constraint implementation. The number of polygons and vertices involved in the model where kept to a small number in order to minimize the rendering demands on the processor. The model is rendered in a north, east, down coordinate system. Figure 25 is a wireframe rendering of the quaternion human body model.

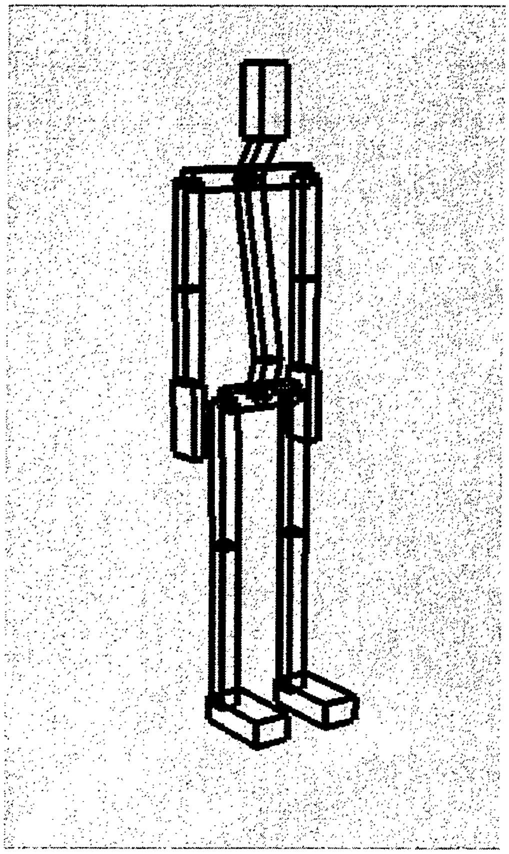
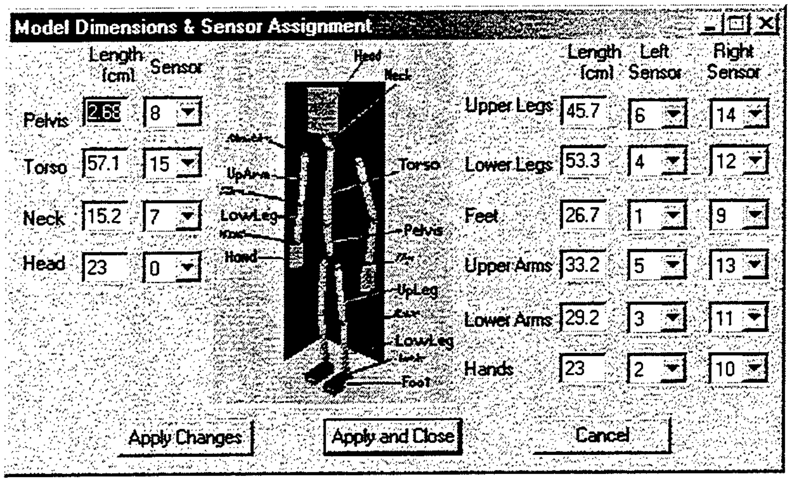


Figure 25: Wireframe Rendering Of The Quaternion-Based Human Model

The orientation quaternions received by the model may be mapped to any number of limb segments. Thus it is possible, depending on the mapping, to animate several limb segments or even the entire model using a single sensor. If one MARG sensor is mapped to all limb segments, the model will rotate as a single unit with an orientation corresponding to that of the applicable sensor. Setting up a one to one correspondence between individual sensors and the movable limb segments of the model would allow realistic tracking and rendering of full body postures. The human model is only a visual approximation of the human body. It is not based detailed studies of human anatomy. The lengths of the individual segments of the model may however be adjusted to match the anthropometric measurements of the individual being tracked. Figure 26 depicts the dialog box used to adjust limb segment lengths and to specify which MARG sensor corresponds to which limb segment or segments. The peculiarities of the manner in which each sensor is attached to each limb segment are accounted for through the use of an *offset quaternion*. The offset quaternions are found using a calibration routine which requires the user to momentarily stand in a reference position. Once the offset quaternions have be calculated, it is assumed that the limb/sensor relationships remain constant.

The human model is implemented in a CHumanModel class. It composed of objects of the CLimbSegment class. CLimbSegment objects encapsulate the length, width, depth, current orientation, offset quaternion and an associated translation vector for each limb segment. Climb segment objects could be used to model any articulated rigid-body. The

CHumanModel class sets the limb segment dimensions and arranges them in a configuration that is recognizable as a human figure. In this research, all limb segments are rendered as a six-sided boxes. To draw figures with a more realistic visual appearance, the limb segment could be extended to include a more complex geometry.



Model Dimensions & Sensor Assignment

	Length [cm]	Sensor
Pelvis	2.68	8
Torso	57.1	15
Neck	15.2	7
Head	23	0

	Length [cm]	Left Sensor	Right Sensor
Upper Legs	45.7	6	14
Lower Legs	53.3	4	12
Feet	26.7	1	9
Upper Arms	33.2	5	13
Lower Arms	29.2	3	11
Hands	23	2	10

Figure 26: Human Model Settings Dialog

a. Setting Model Position and Posture

The vertices of an individual limb segment are described relative to a z-axis down coordinate system which is attached to the inboard end of the segment. If the sensor and limb segment axes are aligned, the orientation of an individual limb segment could be set by applying to each vertex, v, the quaternion rotation

$$q_{sensor} v \bar{q}_{sensor} \tag{6.7}$$

where the unit quaternion q_{sensor} is the estimated orientation produced by the filter processing the sensor output data. In practice, due to the irregular shape of human limb segments and other factors related to sensor mounting and attachment, it is difficult to achieve perfect alignment between the sensor and limb segment axes. This misalignment can be taken into account by performing an additional rotation using an offset quaternion. The orientation of an individual limb segment must then be set by applying the rotation sequence

$$q_{sensor}q_{off}v\bar{q}_{off}\bar{q}_{sensor} \tag{6.8}$$

to each vertex, where q_{off} is the offset quaternion for the limb of the vertex.

To set the position of a individual limb segment it is necessary to find a vector which describes the location of the inboard end of the limb segment. Once this vector is found the final position of each vertex can be calculated through addition of this vector to the rotated coordinates of each vertex. Thus, the final position of a limb segment vertex is given by

$$p_{trans} + q_{sensor}q_{off}v\bar{q}_{off}\bar{q}_{sensor} \tag{6.9}$$

where p_{trans} is a 3-space vector describing the location of inboard end of the limb. Using homogeneous transformation matrices this final positioning could be accomplished by

$$TR_{sensor}R_{offset}v \tag{6.10}$$

where T is a homogenous transformation matrix describing the same translation as v_{trans}, R_{sensor} describes the orientation of the sensor relative to an earth-fixed reference frame and R_{offset} describes the same relation as q_{off}. However, this calculation would be less efficient and is not used in this research.

The *origin* of the human body model is the *waist*. The position of the human model could be set by tracking this location on the user and equating the resulting position vector to the origin. (No position tracking is included in this research.) Attached to the origin are the *torso* limb segment extending generally upward and the *pelvis* limb segment with its long axis orientated in a downward direction when the figure is in a normal standing position. Attached to the outboard end of the torso are the *neck* to which the *head* is attached and the shoulders which have a fixed relation to the torso. The outboard ends of the shoulders are connected to the *upper arms*, to which are attached the *lower arms* and finally the *hands*. The *hips*, *upper legs*, *lower legs* and *feet* are connected to the pelvis in a similar manner.

Each limb segment has an associated translation vector, p, which extends from the inboard to the outboard end of the segment. Once this vector has been oriented using Eq. (6.8) the outboard end point can be used as the origin location for the coordinate system of more distal segment attached to the end point. Limb segment origin positions are calculated through the addition of translation vectors working from the waist towards the body extremities as depicted in Figure 27. Each node represents a limb segment origin and each edge a translation vector which has been rotated by an offset quaternion and a limb orientation quaternion. Positions are determined by traversing the tree from the root to the node of a particular limb segment origin and adding the vectors associated with each edge in the path. For example, by traversing the displayed tree, it can be seen that the elbow or connection point for the inboard end of the right lower arm limb segment is given by

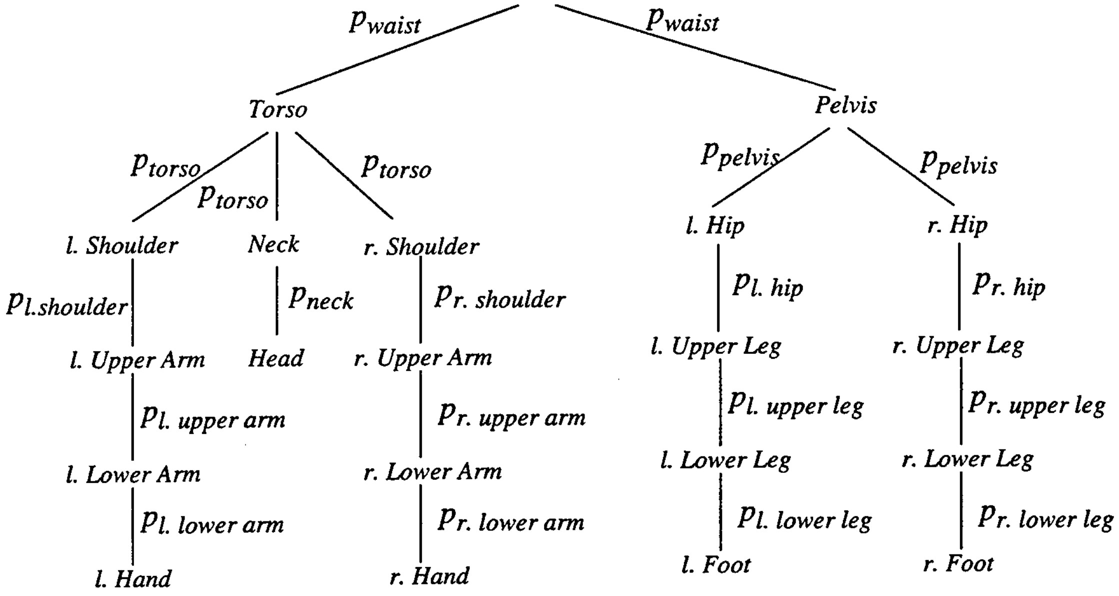


Figure 27: Calculation Of Limb Segment Positions

$$
\begin{aligned}
&p_{waist} + q_{torso} q_{torso\ off} p_{torso} \bar{q}_{torso\ off} \bar{q}_{torso} + \\
&q_{r\ shoulder} q_{r\ shoulder\ off} p_{r\ shoulder} \bar{q}_{r\ shoulder\ off} \bar{q}_{r\ shoulder} + \\
&q_{r\ up\ arm} q_{r\ up\ arm\ off} p_{r\ up\ arm} \bar{q}_{r\ up\ arm\ off} \bar{q}_{r\ up\ arm}
\end{aligned}
\tag{6.11}
$$

in the manner similar to Eq. (3.79). The origins for all other links are located iteratively using the same method.

Actual positioning of the human model is carried out as a two step process. This separation allows rendering calculations to be spread over a wider time interval and thus reduces the impact on the data filtering processes. In the first step the orientations of all limb segments are set. This is accomplished by calling the *setPosture* method of the CHumanModelClass. Filter produced orientations in quaternion form are passed in a predetermined order as an input argument. This method is listed in Figure 28. The limb segments are positioned and rendered using the *renderFigure* method. The location of the waist or the origin of the human figure is passed as input argument. This vector as are all vectors in the software is stored as a quaternion with the real part equal to zero. The renderFigure method is listed in Figure 29.

```
// Sets the orientation for each limb segment
void CHumanModel::SetPosture(CLimbData * angleData)
{
    for (int i = 0; i < 16; i++) {

        // Set the orientation of the limb
        m_trackedLimb[i]->SetOrientation(angleData[i].orientation);

    } // end for

    // Set the orientations of the fixed segments
    m_fixedLimb[L_HIP]->SetOrientation(angleData[PELVIS].orientation);
    m_fixedLimb[R_HIP]->SetOrientation(angleData[PELVIS].orientation);
    m_fixedLimb[L_SHOULDER]->SetOrientation(angleData[TORSO].orientation);
    m_fixedLimb[R_SHOULDER]->SetOrientation(angleData[TORSO].orientation);

} // end SetPosture
```

Figure 28: The setPosture Method Of the CHumanModel Class

b. Body Model Calibration

Due to the irregular shape of actual human limb segments, it is not possible to exactly align the axes of the attached a sensor with those of the limb. Sensor attachment will vary from limb to limb and from individual to individual. Thus the use of off-line

```
// Calls draw functions to draw the human
void CHumanModel::RenderFigure(const Quaternion & vecRoot)
{
    // Calculate joint locations
    Quaternion waist = vecRoot;

    // Calculate upper body joint location vectors
    Quaternion bodyNeck = waist + m_trackedLimb[TORSO]->GetTranslation();
    Quaternion bodyHead = bodyNeck + m_trackedLimb[NECK]->GetTranslation();
    Quaternion lShoulder = bodyNeck + m_fixedLimb[L_SHOULDER]->GetTranslation();
    Quaternion rShoulder = bodyNeck + m_fixedLimb[R_SHOULDER]->GetTranslation();
    Quaternion lElbow = lShoulder + m_trackedLimb[L_UP_ARM]->GetTranslation();
    Quaternion rElbow = rShoulder + m_trackedLimb[R_UP_ARM]->GetTranslation();
    Quaternion lWrist = lElbow + m_trackedLimb[L_LOW_ARM]->GetTranslation();
    Quaternion rWrist = rElbow + m_trackedLimb[R_LOW_ARM]->GetTranslation();

    // Calculate lower body joint location vectors
    Quaternion bodyHip = waist + m_trackedLimb[PELVIS]->GetTranslation();
    Quaternion lHip = bodyHip + m_fixedLimb[L_HIP]->GetTranslation();
    Quaternion rHip = bodyHip + m_fixedLimb[R_HIP]->GetTranslation();
    Quaternion lKnee = lHip + m_trackedLimb[L_UP_LEG]->GetTranslation();
    Quaternion rKnee = rHip + m_trackedLimb[R_UP_LEG]->GetTranslation();
    Quaternion lAnkle = lKnee + m_trackedLimb[L_LOW_LEG]->GetTranslation();
    Quaternion rAnkle = rKnee + m_trackedLimb[R_LOW_LEG]->GetTranslation();

    // Draw the upper body
    m_trackedLimb[TORSO]->Draw(waist);
    m_trackedLimb[NECK]->Draw(bodyNeck);
    m_trackedLimb[HEAD]->Draw(bodyHead);
    // Draw shoulders
    m_fixedLimb[L_SHOULDER]->Draw(bodyNeck);
    m_fixedLimb[R_SHOULDER]->Draw(bodyNeck);
    // Draw upper arms
    m_trackedLimb[L_UP_ARM]->Draw(lShoulder);
    m_trackedLimb[R_UP_ARM]->Draw(rShoulder);
    // Draw lower arms
    m_trackedLimb[L_LOW_ARM]->Draw(lElbow);
    m_trackedLimb[R_LOW_ARM]->Draw(rElbow);
    // Draw hands
    m_trackedLimb[L_HAND]->Draw(lWrist);
    m_trackedLimb[R_HAND]->Draw(rWrist);

    // Draw the lower body
    m_trackedLimb[PELVIS]->Draw(waist);
    // Draw hips
    m_fixedLimb[L_HIP]->Draw(bodyHip);
    m_fixedLimb[R_HIP]->Draw(bodyHip);
    // Draw uppper legs
    m_trackedLimb[L_UP_LEG]->Draw(lHip);
    m_trackedLimb[R_UP_LEG]->Draw(rHip);
    // Draw lower legs
    m_trackedLimb[L_LOW_LEG]->Draw(lKnee);
    m_trackedLimb[R_LOW_LEG]->Draw(rKnee);
    // Draw feet
    m_trackedLimb[L_FOOT]->Draw(lAnkle);
    m_trackedLimb[R_FOOT]->Draw(rAnkle);

} // end RenderFigure
```

Figure 29: The renderFigure Method Of the CHumanModel Class

analytic calculations is not a practical method of deriving an offset quaternion which accounts for the misalignment of the two coordinate systems.

When the human model is in the *reference position*, the limb segment coordinate axes are aligned with the corresponding Earth-fixed axes. That is the x-axis for each limb segment points toward the local north, the y-axis points east and the z-axis points down. The reference position for the human model is an "attention" type stance facing North. The offset quaternions for each limb segment can be derived by noting that while the user is in the reference position the equation

$$v = q_{sensor} q_{off} v \bar{q}_{off} \bar{q}_{sensor} \tag{6.12}$$

is true. This implies that

$$q_{sensor} q_{off} = 1 \tag{6.13}$$

and

$$\bar{q}_{off} \bar{q}_{sensor} = 1 \tag{6.14}$$

These results and the inverse property of quaternion multiplication further imply that

$$q_{off} = \bar{q}_{sensor} \tag{6.15}$$

while in the reference position. The quaternion, q_{sensor}, is output by the quaternion filter algorithm and is thus known.

Complete compensation for the way in which all sensors are attached to the limbs of a tracked subject can therefore be accomplished by simply setting q_{off} for each limb segment to the inverse of the associated q_{sensor} while the subject to be tracked is standing in a predetermined reference position. The implemented reference position for this research is an attention type stance facing the local magnetic north (Figure 30). The calculated offset quaternion will remain valid as long as the sensor positions do not shift position relative to the tracked limb segment.

The same calibration method could be applied to a model based upon the use of homogeneous transformation matrices vice quaternions. Relations which are equivalent to those given by Eq. (6.12) and Eq. (6.15) are

$$v = R_{sensor} R_{off} v \tag{6.16}$$

and

$$R_{off} = R^{-1}{}_{sensor} \tag{6.17}$$

where R_{sensor} is a homogeneous matrix expressing the limb segment orientation relative to a earth-fixed reference frame and R_{off} expresses the orientation of the limb segment coordinate system relative to the that of the sensor. Again, since matrix inversion is very expensive computationally in comparison to unit quaternion inversion, Eq. (6.17) is not used in this research.

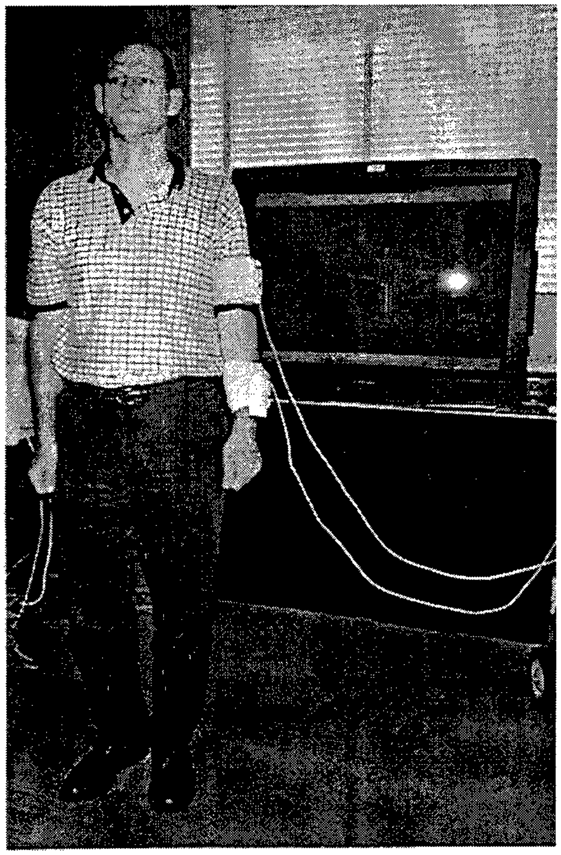


Figure 30: Body Model Calibration Reference Position

D. SUMMARY

The body tracking system described in this chapter is able to track the orientation of human limb segments using prototype MARG sensors. The MARG sensors are calibrated via a series of eight rotations without the need for any specialized equipment. The data from each sensor is processed by a reduced order quaternion attitude filter which is complementary in form. The incorporated filter algorithm is able to track limb segment attitude through all orientations without singularities and continuously corrects for drift. Filter output is a quaternion representation of the orientation of a limb segment relative to an earth fixed reference frame.

The orientation quaternions are used to set the posture of a quaternion based human model. All model segments are positioned and oriented using quaternion/vector pairs in a *z*-axis down coordinate system. The human body model implements a simple calibration method for correcting for misalignment between the coordinate systems of individual sensors and limb segments. The calibration method only involves the inversion and assignment of a single quaternion for each limb segment while the tracked subject stands in a reference position. The minimal computational demands of this method are largely due to the overall simplicity of the human model itself. Human model limb segment lengths may be adjusted to account for human anthropomorphic differences.

The prototype MARG sensors where fabricated using low-cost off-the-shelf components. Internal sensor circuitry supports magnetometer set/reset of MR film polarity and allows manual adjustment of magnetometer null points. Analog MARG sensor output digital conversion is performed external to the sensors using a PCI data acquisition card.

The system software was entirely implemented using C++. It is single threaded and runs on a standard Wintel desktop computer. The estimation update rate is at least 100 Hz for three filters performing drift correction calculations on each iteration. Rendering frame rate is maintained at 25 Hz.

VII. EXPERIMENTAL RESULTS

A. INTRODUCTION

The chapter describes experiments designed to test the performance of the inertial/ magnetic body tracking system. The quantitative and qualitative results presented document the accuracy and robustness of the system under various dynamic and static conditions.

The static experiments described relate to the stability, convergence properties and accuracy of the orientation estimates produced by the quaternion attitude filter algorithm when processing MARG sensor data. All static tests were single MARG sensor experiments. Preliminary results are presented which quantitatively illustrate the dynamic accuracy of the quaternion filter orientations. This data also allows some conjectures to be made regarding system latency.

The qualitative experiments examine the performance of the system as a whole in relationship to the goal of robust posture estimation. The performance of the system while using differential weighting of sensor data as well as increased drift correction intervals is investigated. The ability of the system to track the posture of various limb segments of the human body using three MARG sensors is also qualitatively evaluated.

The final section of this chapter examines the InertiaCube sensor and Kalman filter algorithm used by Intersense Inc. to process inertial data [Ref. 27.]. The shortcomings of this system for full body tracking applications are discussed. This discussion is based upon both the observed performance of an Intersense system and an examination of available research literature describing the implemented data filtering algorithm.

B. STATIC STABILITY

Orientation estimates based solely on angular-rate sensors are prone to drift problems. Thus in the past, the idea of using inertial sensors to track orientation for extended periods was often criticized due to the mistaken belief that the estimates would diverge over time. This criticism was applied regardless of the combination of sensors actually in use and was mostly due to difficulties in understanding the complementary characteristics of different sensor types and complementary estimation filters.

The drift characteristics of the quaternion filter algorithm and the MARG sensor over extended periods were evaluated using static tests. In each of these experiments the stability of the orientation estimate produced with the sensor stationary was monitored for a specified period. Through the course of the experiments the estimated orientation was recorded at 0.1 second intervals. Figures 31 through 34 display the results. Each plots the four components of the estimated quaternion and the length of the quaternion error vector versus time.

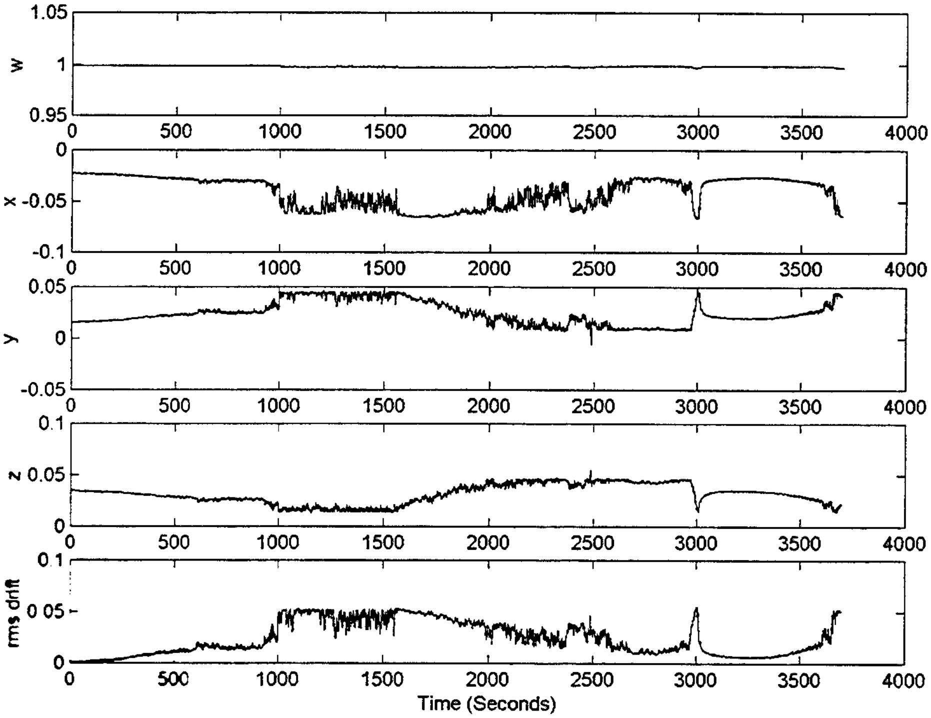


Figure 31: One Hour Static Test Of Orientation Estimate Stability, k = 1.0, ρ = 1.0

Figure 31 graphically depicts the drift of each of the four components of the quaternion estimate produced by the filter. It can be observed through examination of Figure 31 that average total drift is about 1%. During the experiment shown, the filter gain, k (Eq. (5.15)), was set to unity. Equal weighting was given to both magnetometer and

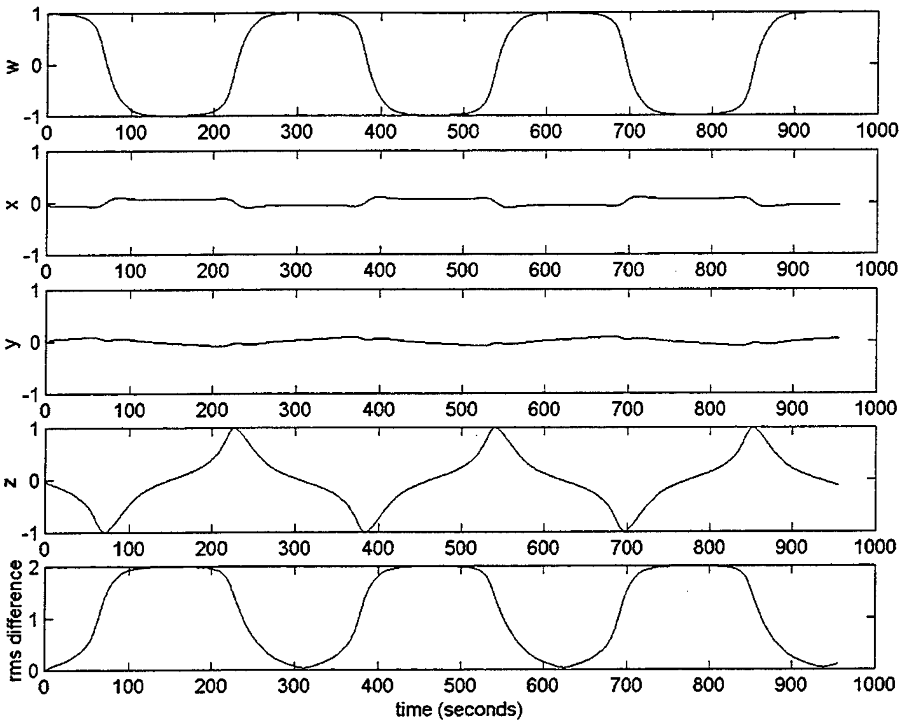


Figure 32: 15 Minute Static Test Of Orientation Estimate Stability, No Magnetometer Input, *k* = 1.0

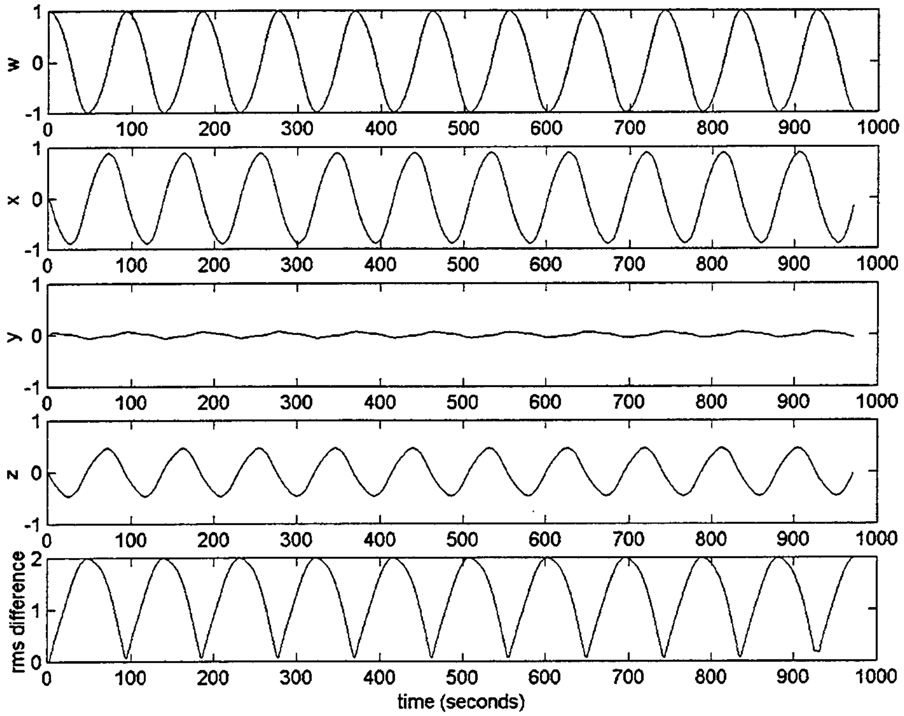


Figure 33: 15 Minute Static Test Of Orientation Estimate Stability, No Accelerometer Input, *k* = 1.0

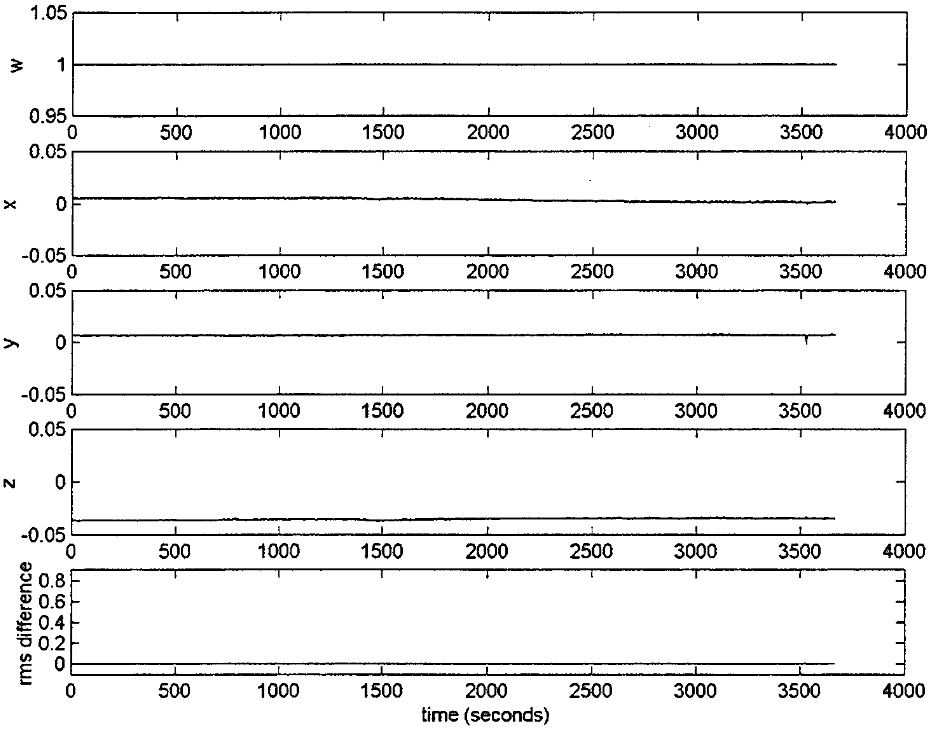


Figure 34: 60 Minute Static Test Of Orientation Estimate Stability, No Rate Sensor Input, *k* = 1.0

accelerometer data. It is expected that increasing the filter gain to 4.0 would reduce the drift error by a factor of four or to about 0.25 percent. However, due to the observed stability of the filter over a one hour period, no further static experiments relating to stability were conducted.

Experiments were also conducted in which magnetometer, accelerometer or rate sensor data were disregarded by the filtering algorithm. These results are shown in figures 32, 33 and 34. As expected, Figure 32 shows continuous drift about the vertical axis of approximately 1 degree per second. Poor stability about the North and East axes is apparent in Figure 33. Here the total drift is on the order of 3 degrees per second. The greatest possible difference between two unit quaternions occurs when they point in directions which are exactly opposite each other. At that time the length of the error vector would be two. Thus, the magnitude of the rms difference in both Figure 32 and Figure 33 cycles between 0 and 2.

The lack of any drift in Figure 34 indicates that all drift is due to rate sensor bias. Throughout these tests all limb segments of the human model were mapped to the single sensor in use. The posture of the human model when viewed on the display monitor reflected the results seen in the figures.

C. STATIC CONVERGENCE

Linear analysis of the quaternion filter and nonlinear simulation imply the transient errors in $\hat{q}$ will persist for a time period which is inversely proportional to the filter gain, k (Eq. (5.15)). Specifically, by the time $\tau_{\Delta q}$ following the occurrence of a transient error, it is expected that the error magnitude will be reduced to 37% of its original value (Eq.). The magnitude of the square of the error should be reduced by 37% by the time

$$t = \frac{\tau_{\Delta q}}{2} \tag{7.1}$$

Experiments to test the static convergence of the filter following transient errors were conducted to further validate the results of the linear analysis. The MARG sensor itself was left in a stationary position throughout each of these experiments. Transient orientation errors were introduced into the system by rotating a stable $\hat{q}$ estimate by an *error quaternion*. Following this rotation, the filter was allowed to reconverge to the previous estimate. Error quaternions representing orientation errors of 30 degrees where used. Filter gains included 1.0, 4.0, 8.0, 16.0 and 32.0. Setting the filter gain to values greater than 200 with an update rate of 100 Hz (as predicted by Eq. (5.44)) was found to make the filter unstable. Equal weighting was given to both magnetometer and accelerometer data. In each of these experimental trials the filter remained stable and re-converged to the previous estimate in the time period predicted by linear theory. Figures 35 through 38 plot the magnitude of the quaternion filter criterion function (Eq. (5.10)) versus time. These data were obtained following rotation of $\hat{q}$ by the indicated error quaternion. Filter gains of 1.0, 4.0, 16.0 and 32.0 are shown. These figures represent a sample of the results obtained.

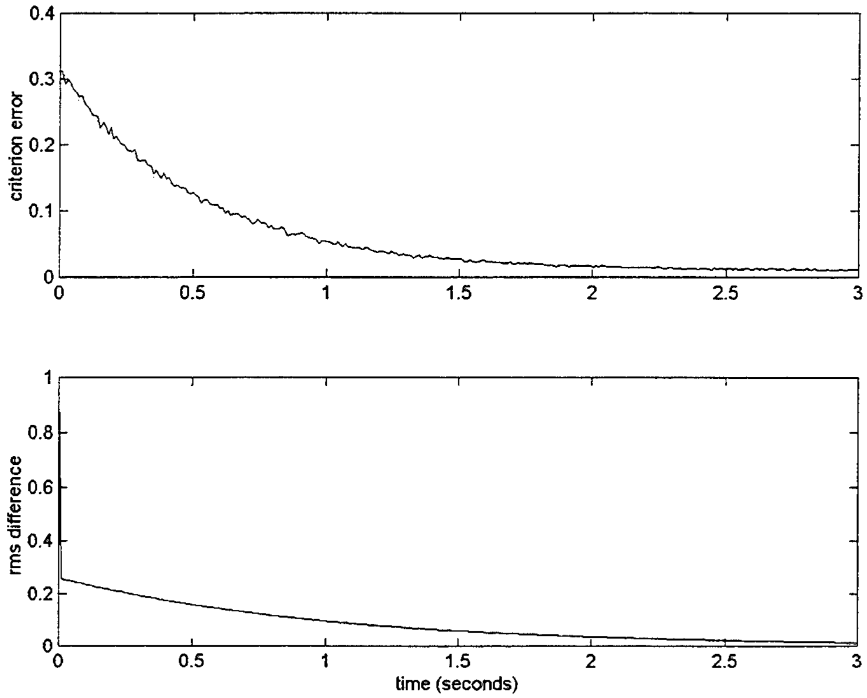


Figure 35: Error Convergence Following 30 Degree Transient Error, *k* = 1.0

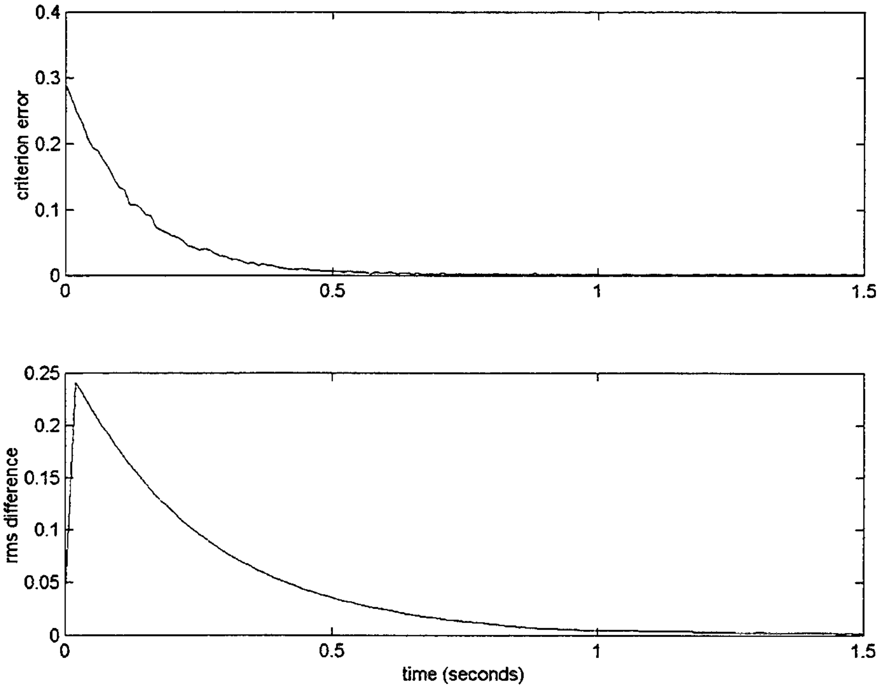


Figure 36: Error Convergence Following 30 Degree Transient Error, *k* = 4.0

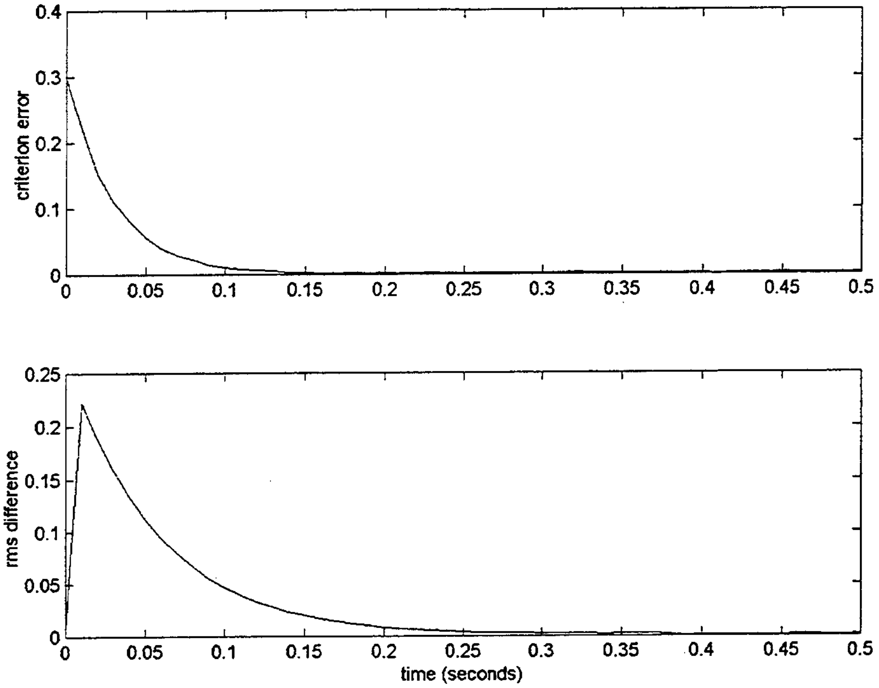


Figure 37: Error Convergence Following 30 Degree Transient Error, *k* = 16.0

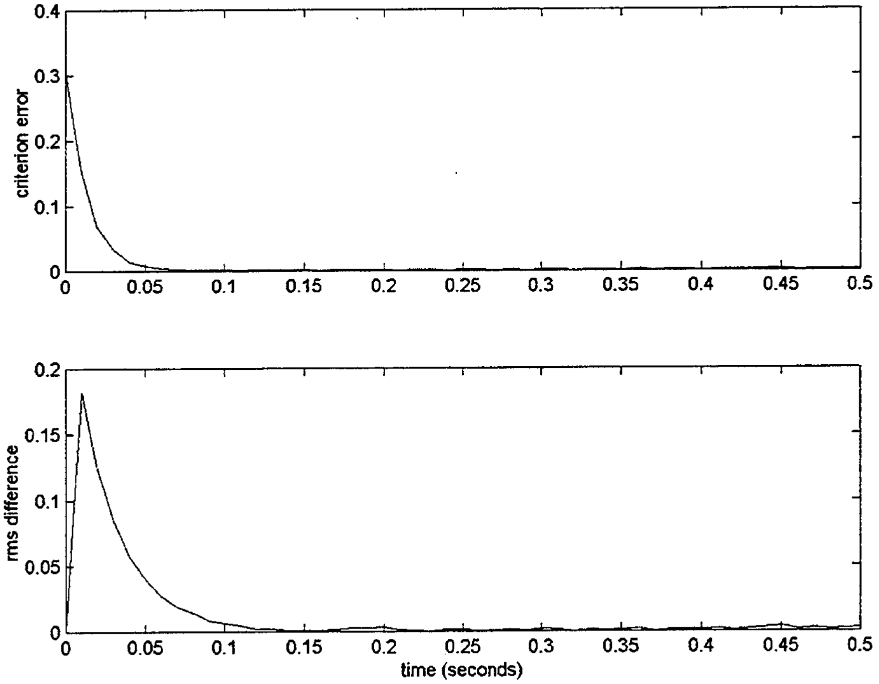


Figure 38: Error Convergence Following 30 Degree Transient Error, *k* = 32.0

D. DYNAMIC RESPONSE AND ACCURACY

Preliminary experiments were conducted to establish the accuracy of the orientation estimates and the dynamic response of the system [Ref. 6.]. These experiments were completed using a Hass rotary tilt table [Ref. 31.]. The table has two degrees of freedom and is capable of positioning to an accuracy of 0.001 degrees at rates ranging from 0.001 to 80 degrees/second. In order to mitigate any possible magnetic field effects generated by the servos of the tilt table, the sensor package was mounted on a nonferrous extension above the table. The extension was approximately the length of a human forearm.

The preliminary test procedure consisted of repeatedly cycling the sensor through various angles of roll, pitch and yaw at rates ranging from 10 to 30 deg./sec. After each motion, the table was left static for approximately 15 seconds. The estimates produced by the tracking system where converted to Euler angle form for easier comparison to the tilt table rotations.

Figure 39 is a typical result of the dynamic accuracy experiments. The overall smoothness of the plot shows excellent dynamic response. Accuracy was measured to be better than one degree. The small impulses which can be observed each time motion is initiated are hypothesized to be linear acceleration effects exaggerated by the "whipping" motion of the extension on which the sensor was mounted. In qualitative tests, the quaternion filter exhibited no difficulty in tracking orientations in which pitch angles equaled or exceeded 90 degrees.

E. QUALITATIVE TESTING

1. Weighted Least Squares

The weighted least squares modification to the quaternion filter algorithm is designed to allow orientation estimation to continue in the presence of changing magnetic fields. Significantly reducing the weight given to magnetometer data will allow drift about the vertical axis. However, this reduction may also make it possible to avoid large short time period rotations about the vertical axis in the presence of changing magnetic fields.

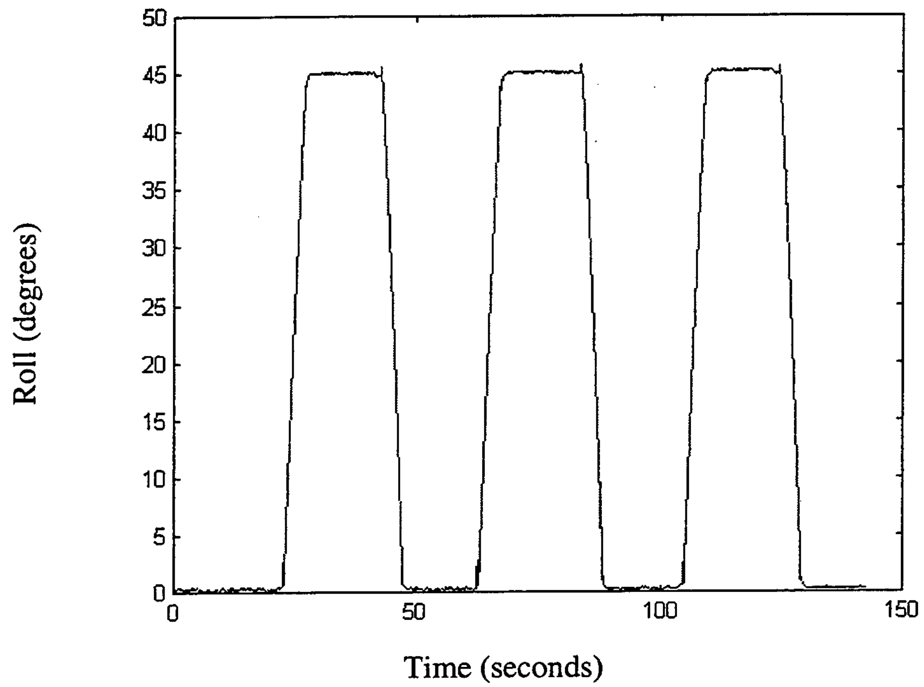


Figure 39: 10 Degree Roll Excursions At 10 deg/sec From [Ref. 6.]

To examine the weighted least square function of the filter, a MARG sensor was repeatedly subjected to a magnetic source. In each trial a speaker magnet was passed over the sensor at a distance of approximately 1 cm. Magnetometer weighting values of 0.25, 0.5, and 1.0 were used. The filter gain, *k*, was 4.0 in all trials. Figures 40 through 42 plot the rms difference between the orientation estimate before exposure to the field and during exposure. As expected using a magnetometer weighting factor of 0.25 allows the greatest immunity to magnetic field effects as reflected by Figure 42.

2. Posture Estimation

The purpose of the human body tracking system is to estimate the orientation of multiple human limb segments and use the resulting estimates to set the posture of the human body model which is visually displayed. Numerous experiments were conducted to qualitatively evaluate and demonstrate this capability.

In each experiment three MARG sensors where attached to the limb segments to be tracked. Due to the minimal number of sensors available tracking was limited to a single

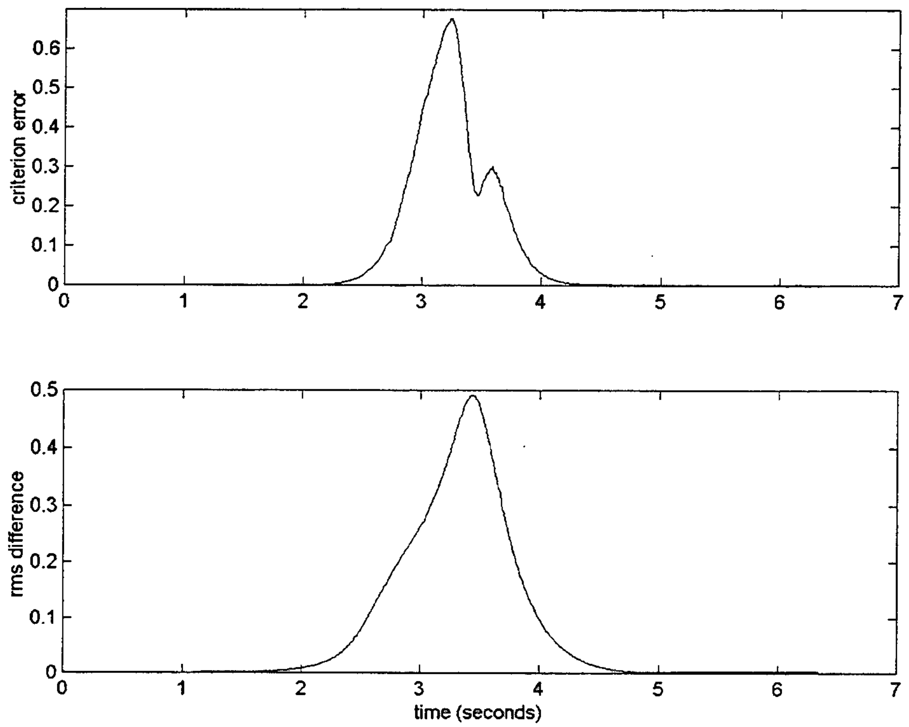


Figure 40: rms Change In Orientation Estimate During Exposure Magnetic Source, Magnetometer Weighting Factor: 1.0, *k* = 4.0

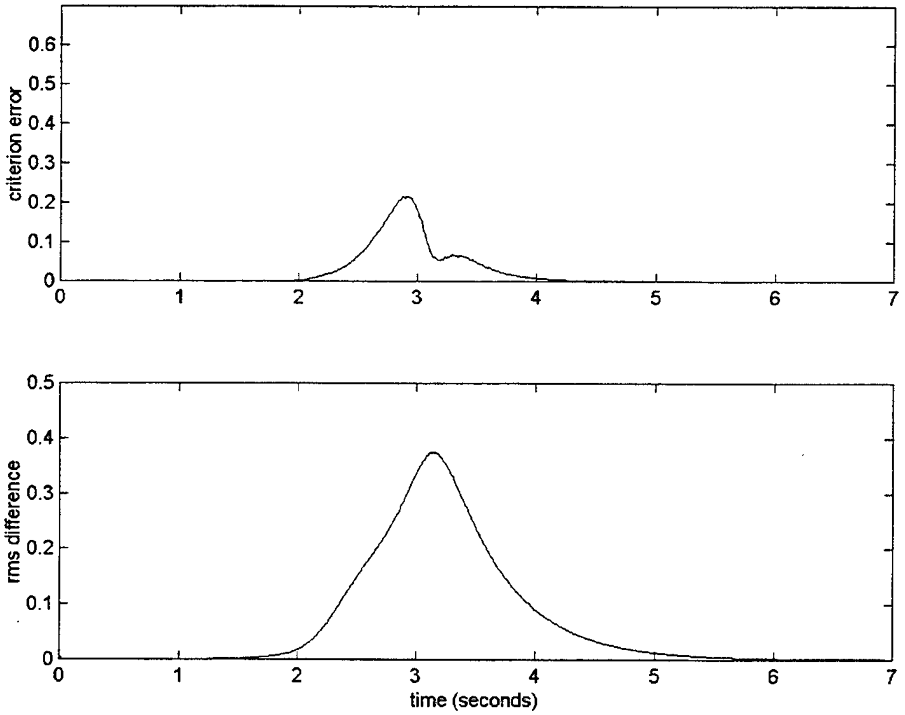


Figure 41: rms Change In Orientation Estimate During Exposure Magnetic Source, Magnetometer Weighting Factor: 0.5, *k* = 4.0

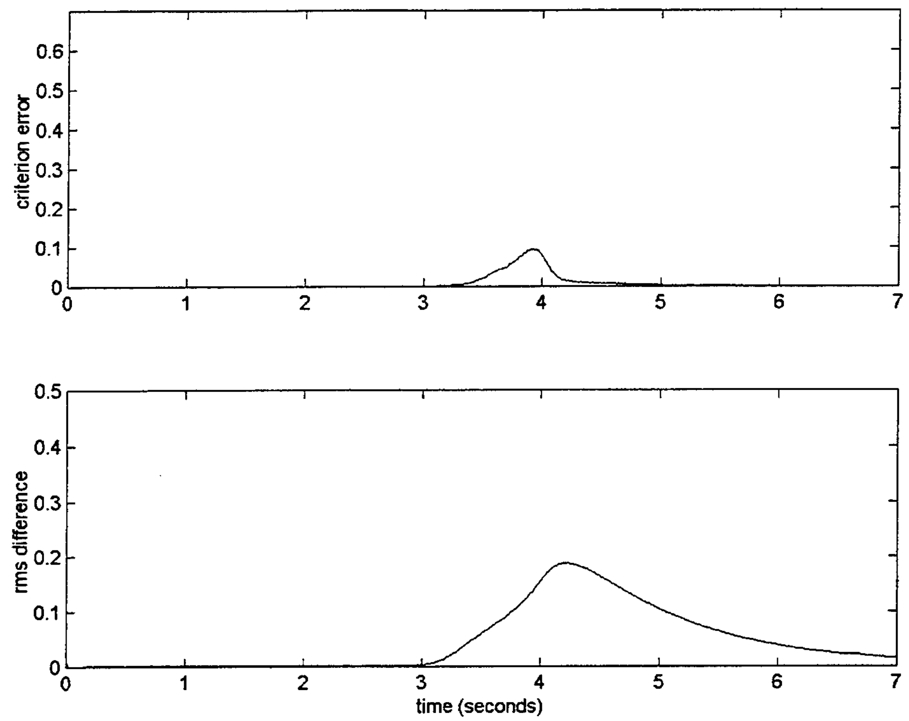


Figure 42: rms Change In Orientation Estimate During Exposure Magnetic Source, Magnetometer Weighting Factor: 0.25, k = 4.0

arm or leg. In the case of arm and limb segments, sensor attachment was achieved through the use of elastic bandages. In most cases this method appeared to keep the sensors fixed relative to the limb. Sensor attachment was the most time consuming task when preparing to track a new individual. Calibration for sensor/limb axes misalignment was the achieved in a nearly instantaneous manner. Adjustment for differences in anthropometric measurements were carried out on an "as needed" basis to allow capture of closed loop postures.

Video recordings of the system in operation indicate that posture estimation was accurate and showed very little lag. Figures 43 through 44 depict inertial tracking of various limb segments.

3. Reduced Rate Drift Correction

It was hypothesized that increasing the drift correction interval for each sensor/filter pair would allow full body tracking without an increase in processing power. To test this

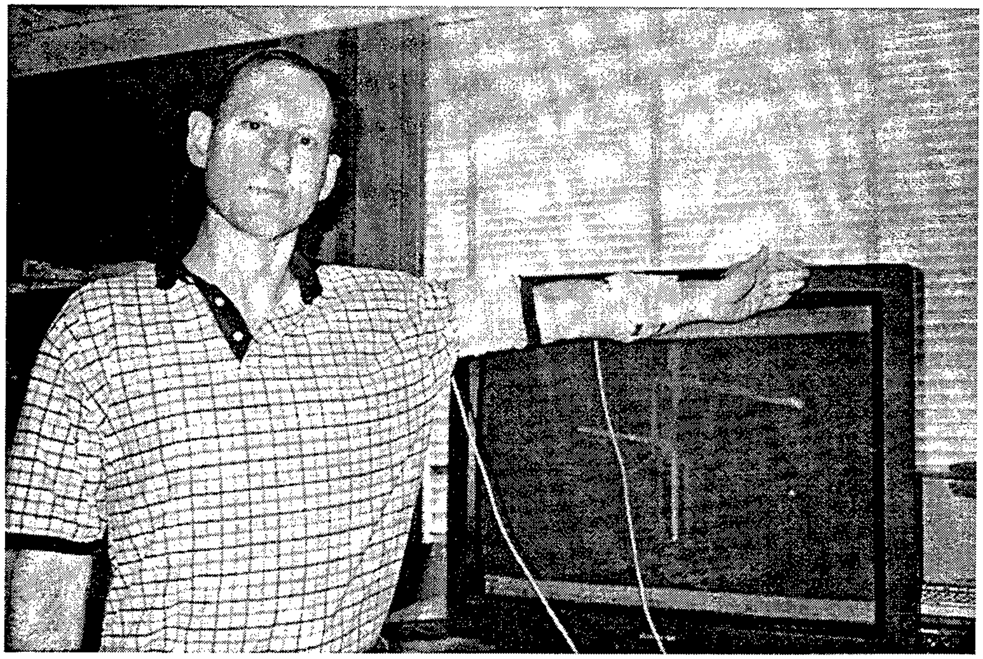


Figure 43: Inertial Tracking Of the Left Arm Using Three MARG Sensors

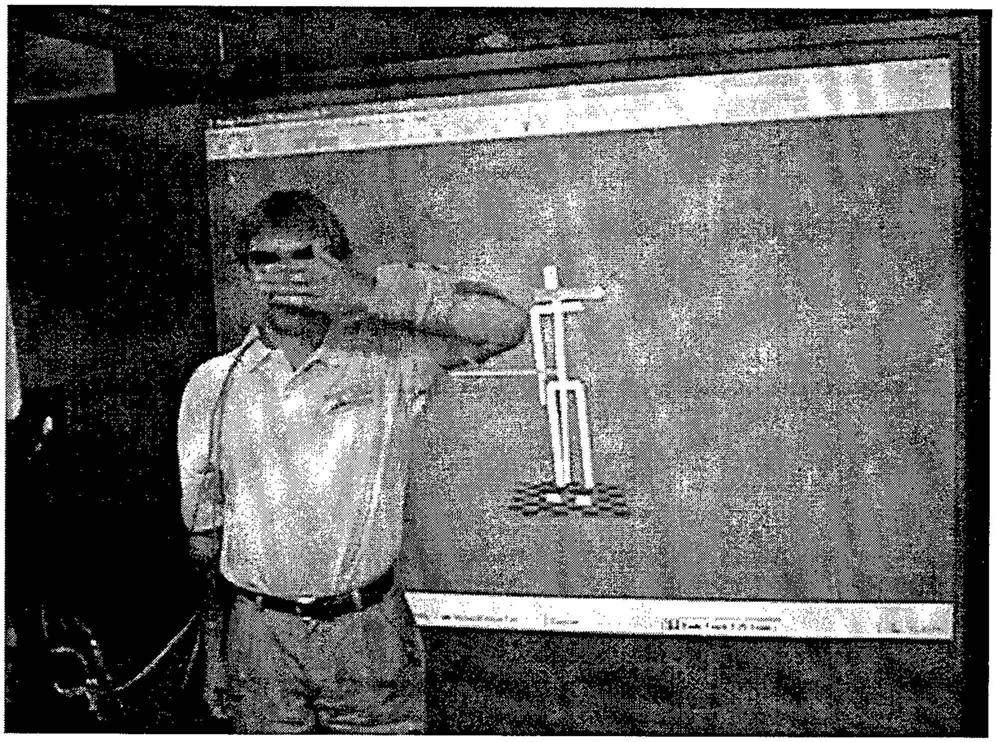


Figure 44: Closed Kinematic Chain Posture Using Three MARG Sensors

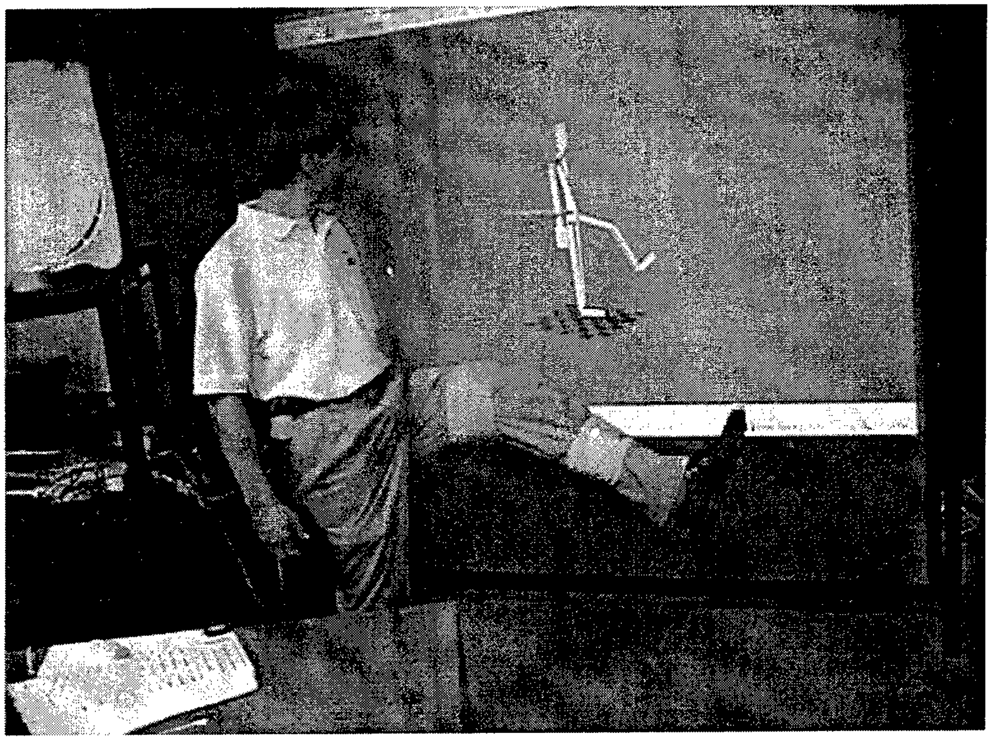


Figure 45: Inertial Tracking Of the Left Leg Using Three MARG Sensors

hypothesis, posture estimation was evaluated qualitatively while operating three actual and 13 simulated filter software objects. Filter update rates were maintained at 100 Hz. However, only one filter object performed drift correction calculations on each update cycle. This is equivalent to performing drift corrections for each filter object at a rate of approximately 6 Hz. Filter gains where not changed to compensate since the most recently derived drift correction factor was still used on every update cycle. Qualitative evaluation of posture tracking indicated the effects of increasing the drift correction interval for each filter object were negligible.

F. INTERSENSE INERTIACUBE

The InertiaCube (Figure 46) is an integrated inertial sensing device manufactured by InterSense Inc. The InertiaCube senses angular rates about and linear acceleration along each of three orthogonal body axes [Ref. 26.]. Manufacturers literature indicates that it contains at least a two axis magnetometer [Ref. 37.] and thus is very similar in overall capability to the MARG sensors used in this research. InertiaCube data is processed by a complementary separate-bias extended Kalman filter based upon an Euler angle representation of orientation [Ref. 27.]. It has a six-dimensional state vector containing the three Euler angles and three angular rate bias terms. Though the system was designed for head tracking, head dynamics are not modeled. Nonlinear attitude computation is accomplished through a second order integration step formula. System error dynamics are obtained by normalizing about a nominal trajectory [Ref. 27.].

Qualitative evaluation of an Intersense IS-300 orientation tracking system was completed using both the manufacturers demonstration software and the body tracking software developed for this research. When using the demonstration application the basis of the filter in Euler angles becomes apparent each time the InertiaCube is subjected to a pitch angle approaching +/- 90 degrees. In this attitude the roll and yaw values gyrate widely while maintaining a constant sum or difference. When tested with the body tracking software, the system was configured to output a quaternion representation of orientation. While operating in this mode it was not possible to detect any singularities.

The ability of the InertiaCube to continuously correct for rate sensor drift was tested by subjecting the sensor to a series of accelerations and then placing it on a flat surface. When using either the demonstration application or the body tracking software, the system exhibited its inability to correct for drift unless in a stationary state. Each time the sensor was replaced on the flat surface the orientation estimate failed to match the true orientation for a short time period before making a sudden and abrupt correction. This phenomenon occurred regardless of the operating mode of the Intersense system and is in marked contrast to continuously corrected estimates produced by the MARG sensor and the quaternion attitude filter.

The strength of Kalman filtering lies in the inclusion of a dynamics model for error correction and prediction. In the absence of an accurate model, use of a Kalman filter is likely to result in an unnecessarily complex algorithm which is prone to errors when the model does not match the dynamics of the modeled system [Ref. 14.]. Since Kalman filter predictions are primarily based upon dynamics models, any prediction produced by a model-less filter should be viewed with suspicion. The inclusion of adjustable gains to control rms estimation error and attenuate magnetometer inputs raises doubts that the Intersense system is actually using a Kalman gain to attain a statistically optimal estimate of the system state. It appears instead that the system may implement a highly expensive extended Kalman filter algorithm to perform a task which could be done quicker and faster by a simpler algorithm such as the quaternion filter algorithm described in this research.

While the inability of the Intersense system to continuously correct for drift may not be a drawback in head tracking applications, it is doubtful that it will be able to function properly in a constant high acceleration applications such as full body tracking. There are no periods of still time for the limbs of a human being while walking. Though no singularities were observed in the experiments described here, the use of Euler angles to describe the orientation of a human arm which can assume any attitude is questionable. The decision to date of Intersense not to allow direct access to the signals produced by the InertiaCube sensor severely limits the application of inertial tracking technology with this sensor.

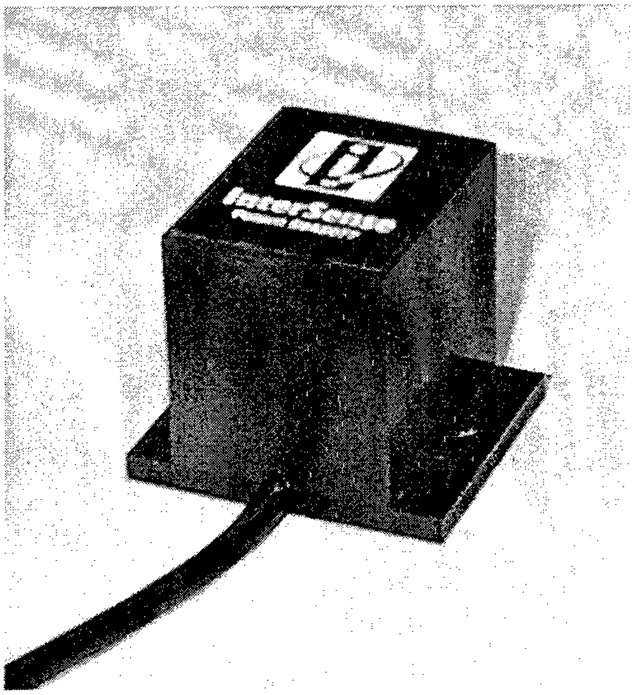


Figure 46: Intersense InertiaCube

G. SUMMARY

This chapter presents a limited set of experiments designed to document the performance parameters of a prototype inertial/magnetic body tracking system. The accuracy of system orientation estimates was quantitatively evaluated both statically and dynamically. The overall ability of the system to model human posture in real-time was evaluated qualitatively. Examination in the light of human body tracking was also made of the Intersense InertiaCube and filtering algorithms.

The experimental results presented indicate that with the application of the proper algorithms and representations, inertial/magnetic orientation tracking can be used to accurately track the posture of the human body. The static stability and convergence tests show that the orientation estimates are stable and that linear analysis of quaternion filter analysis is valid. The static accuracy and dynamic response experiments show the system can produce orientation estimates which are accurate and timely enough to be used in real-time body tracking applications. For applications in which the system may be subjected to variable magnetic fields, the weighted least square experiments show that inertial/magnetic tracking may still be expected to produce usable orientation estimates. Qualitative experiments show increased drift correction intervals may be used to implement a tracking system which operates a larger number of filters simultaneously using limited processing.

Video recordings of the system in operation demonstrate that inertial/magnetic orientation estimation produces accurate body posture estimates in real-time. Sensor and body model calibration algorithms make the technology robust and easy to use. With the addition of a wireless link and an appropriate position tracking technology, it is apparent that the prototype system represents a means of simultaneously tracking a large number of users in a large work area without the shortcomings of current motion capture technologies.

Experimentation with gains and scale factors makes is apparent that it is useful to think of the rate sensor data as primarily serving to "quicken" the orientation estimates produced using accelerometer and magnetometer data. In head tracking applications this may be necessary in order to reduce lag and avoid the possibility of simulator sickness. Quickening may also be needed in feedback control applications to ensure stability. However, in some body tracking applications it may be possible to use simpler sensors including only magnetometers and accelerometers with a relatively high filter gain. This possibility presents an important area for future research.

VIII. SUMMARY AND CONCLUSIONS

A. INTRODUCTION

This research lays the groundwork for a system capable of sourceless tracking of the entire human body. The technology of such a system should allow the tracking of multiple users over a wide area. Ultimately, each user could be inserted into a networked synthetic environment in a fully immersive manner. While the basic ground work and theory have been completed in this dissertation, much research remains to complete a full body tracking system. Numerous technologies must be merged and adapted to produce a practical body tracking system for networked synthetic environment applications.

The following outlines the work which needs to be done to achieve full body tracking and makes suggestions regarding what directions this work should take. The implications of this research are discussed. The final section of this document examines what conclusions might be drawn from its contents.

B. MARG SENSORS

An optimal inertial sensor would have the same size and form factor as a wristwatch. It would include an embedded microprocessor on which the filter algorithm is implemented. The sensor would be have a self-contained power source and would wirelessly transmit orientation data.

New sensor components continue to appear on the market. These sensors have capabilities which are at least equal to and are often superior to those of the preceding generation and are an order of magnitude smaller in size. Current technology already permits the construction of sensors which are much smaller than either the prototype MARG sensors described here or the InterSense InertiaCube. Honeywell now offers the HCM1023 three-axis magnetoresistive sensor in a sixteen pin package with an 8.13 x 3.81 mm footprint. This unit is less than half the size of the HMC2003 with a nearly equal sensitivity [Ref. 38.]. The Analog Devices ADXL202E is a two axis acceleration sensor integrated onto a single monolithic IC chip. It measures 5 x 5 x 2mm and has a digital output [Ref. 2.]. Tokin America Inc. now offers a smaller version of the CG-16D ceramic gyro designated the CG-L33. The CG-L33 measures 8 x 16 x 5mm and has slightly improved performance characteristics [Ref. 83.].

Smaller and cheaper integrated inertial sensors are on the horizon. In the Spring of 2001, it is expected that Micro Sensors, Inc. will release a 5 x 5mm micromachined rate sensor with capabilities equal to those of the CG-L33 [Ref. 63.]. In the Fall of 2000, Tokin America, Inc. released in prototype the MDP-A3U7 3D Motion Sensor [Ref. 82.]. Manufacture's literature states that this sensor unit contains in combination ceramic gyros, acceleration sensors and terrestrial magnetism sensors and is capable of detecting the 3-dimensional posture angle of a body to which it is attached in real time. The sensor measures 25 x 36 x 22.5mm and interfaces via USB (Universal Serial Bus). Maximum errors in all axes are claimed to be +/- 15 degrees. Maximum pitch and roll angles are limited to +/- 60 degrees. It is likely that the large magnitude of the estimation errors as well as the limited pitch and roll capabilities are due to manufacturer's data processing algorithm and are not characteristics of the sensors themselves.

The prototype MARG sensors used in this research output nine analog signals corresponding to the nine sensor axes. The Texas Instruments TLC2543 is an 11 channel, fully configurable, analog to digital converter (ADC) on a single IC. [Ref. 81.] Incoporation of an ADC into the MARG sensor would ease data handling by replacing the 15 wires per MARG sensor with 3 data wires and 2 power wires. The 3 data lines could be merged into a data bus of 15 other MARG sensors. The ADC would also automate the magnetometer set/reset circuit by providing clocked and, therefore, constant readings of set and reset produced magnetic data. Using the difference of the two magnetic readings taken during the set/reset cycle will result in magnetic data that is automatically temperature compensated. This is something that was not possible with the analog MARG sensor used in this research.

In earlier body tracking work, the angular rate sensors were bias compensated in software [Ref. 6.]. In the research described in this document, the hardware is considered stable enough to eliminate the need for these additional calculations. However, integration of a biased angular rate signal will cause a steady state error in a complementary filter. In order to achieve better system performance, this correction should be hardware implemented in the rate sensor conditioning circuitry using capacitive coupling. Such a bias compensation circuit is depicted in Figure 47.

It is expected that use of the components described here would result in a sensor which is as much as five times smaller than the prototype used in this dissertation. Such a sensor would be less expensive, easier to calibrate and mount to a human body and perhaps more accurate as well.

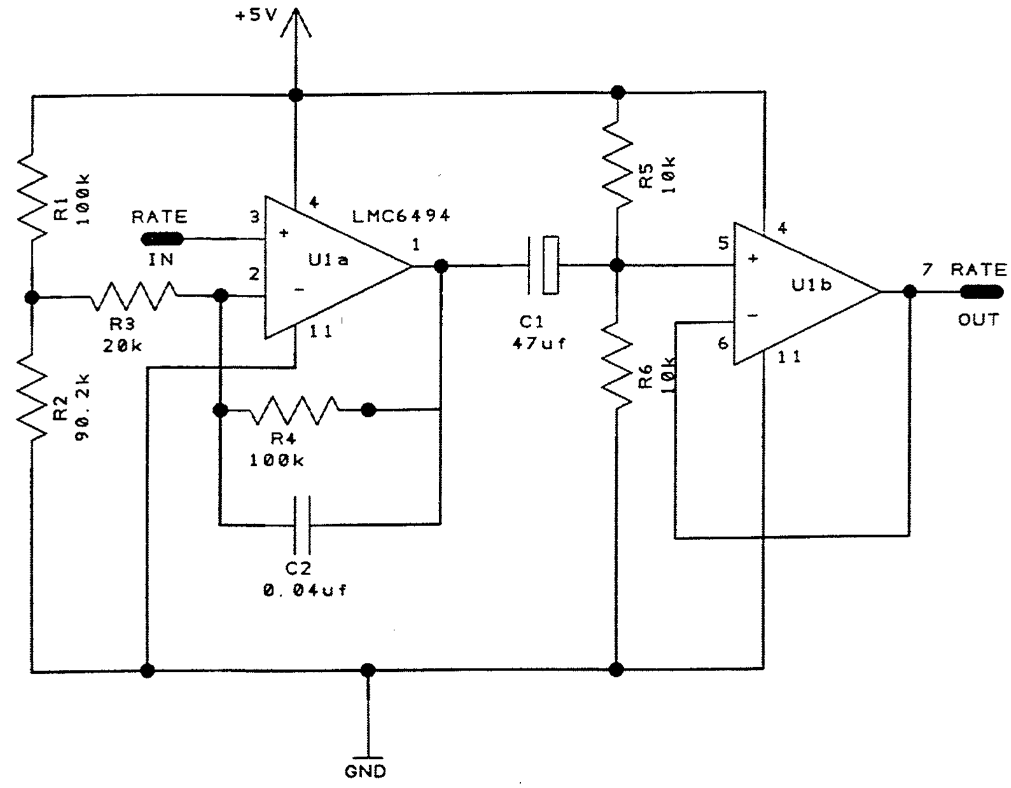


Figure 47: MARG Rate Sensor Bias Compensation Circuit Schematic From [Ref. 61.]

C. HUMAN BODY MODELING

The purpose of the human model used in this research is to demonstrate the simplicity of the mathematics underlying an articulated body model designed to use accurate fixed reference frame orientation data. Speed of rendering and length of development where primary considerations rather than visual appearance.

Future research should expand this paradigm to include anatomical data to make the model correspond more closely to the human skeleton in overall proportion and relative placement and attachment. The model should remain mathematically simple. Due to the accuracy and low cost of inertial/magnetic orientation tracking, there is no need to include joint angle constraints or the ability to track multiple segments using a single sensor. Continuing the direct use of quaternions to orient limb segments and vector addition to position them, may or may not be advantageous depending upon the rendering speed advantages of matrix based graphics hardware and the available network bandwidth.

Current human animation standards model articulated structures using segments and joints [Ref. 33.][Ref. 10.]. Unlike the model used in this research which orients limb segments individually using data referenced to an earth-fixed frame, typical humanoid animation is performed by altering the angle or angles for each individual joint. The orientation of each limb segment is described relative to the inboard segment to which it is attached. Conversion of earth-fixed reference frame data to a series of relative joint angles could be accomplished. However, joint angle animation is actually less efficient than the method used in this research while the network bandwidth requirements are roughly comparable. Thus, joint angle based standards should be expanded to allow this alternate method of setting body posture. Alternatively, efficient routines for converting earth-fixed limb segment orientations to sequential relative joint angles might find wide use.

The current calibration algorithm is effective and easy to use. Once the subject of the tracking experiment stands in a predefined reference posture, sensor to limb segment offset compensation can be accomplished in a time period which appears to be instantaneous. The calibration is based on the assumption that the limb segment coordinate axes are aligned with earth-fixed axes referenced to the local magnetic field. Since it is difficult to have a human subject stand in a very precise pose, in reality there is likely to always be some misalignment between the two coordinate systems. Using the ability of the sensor mounted magnetometers to find the local north will allow the implementation of a two step calibration algorithm. The human subject will still be required to stand in a predefined position, however there will no longer be any requirement to face the local magnetic north. This north finding ability will also make it possible to reduce offset errors further when the subject does face the local north. "Visual tuning" of offsets could be accomplished by "on-screen" adjustment of the displayed posture.

In order to ensure that the user can effectively interact with the virtual environment, the model used by the inertial tracking system must be scaled to the user's dimensions [Ref. 78.]. This type of calibration ensures that, for example when a subject touches their right shoulder with their left fingertips, their virtual human representation will do so as well. Currently, the model is calibrated to body dimension ratios manually through physical measurement. This is an extremely error prone and time consuming process. Some body dimensions such as inseam are easily measured. Using a minimal set of such dimensions it should be possible to accurately calculate other dimensions through a calibration algorithm. The algorithm might involve placing the subject in a series of predefined positions as well as model adjustment based on the rendered posture while in these positions. Such an algorithm would make it possible for a user of any size to easily enter the virtual environment.

At the time of this writing, 3D color laser scanning is being used to digitize the dimensions of recruits at the Marine Corps Recruit Depot in San Diego, CA. The scanning process requires a total of 15 - 20 seconds and produces a detailed anatomical model of each subject [Ref. 19.].This same technique could be used to produce a human model that is perfectly sized for each individual.

There is no end to the amount of effort that could be expended to produce a body model with a realistic appearance. A great deal of research has already been done in this area. Limb segment surfaces could be made anatomically accurate. Muscles and fat could be modeled. Realistic clothing could be added. Hair and facial expressions could be modeled. Individual fingers could be tracked using another technology. In the end, the rendering hardware and the application will dictate how much of this work should be applied to the human body model presented here.

D. INTERGRATION OF INERTIAL AND RF TECHNOLOGIES

The ultimate goal of this project is to insert humans into a networked virtual environment. A network of 15 MARG sensors will track body posture. In order to accurately place the icon of the user in the virtual environment, it will be necessary to know body location as well as the posture of the body. To achieve this, the position of one body limb segment must be tracked. Unlike acoustic position tracking, Radio Frequency (RF) positioning systems are very fast and long range by their nature. Large working volumes can be covered using a minimal amount of equipment and positional error magnitudes remain constant though out. RF positioning systems can penetrate objects, walls, and the human body, and are able to operate with no line-of-sight. Thus, RF positioning is currently seen as the technology which will best complement the sourceless capabilities of inertial/ magnetic sensing and enable tracking of a multiple users over a wide area.

Current examples of RF positioning systems include the Global Positioning System (GPS) and Long Range Navigation (Loran). In an outdoor application where extremely accurate positioning is not required, GPS might be used to locate the position of the tracked subject. DGPS has already been successfully integrated with inertial and magnetic sensors in AUV navigation systems such as the SANS described in [Ref. 7.][Ref. 96.]. Recently, MIT has developed an RF positioning system that shows excellent performance for indoor tracking [Ref. 24.]. This system has an accuracy of 2mm within a range of about 5 meters. For 3 DOF tracking, a minimal system requires four transmitter stations placed at known locations, and a receiver unit attached to the body. Such a system could be easily integrated with a 15 MARG sensor system.

E. WIRELESS COMMUNICATIONS

Tethering a tracked subject to a workstation with wires increases user encumbrance and limits the application of hybrid inertial body tracking technology. The MARG sensors are sourceless sensors, and RF positioning systems are self-contained receiving units. They do not require wire connections to any external sources in order to operate. Thus, a practical hybrid system would incorporate multiple MARG sensors, at least one RF position tracker and a wearable electronics unit capable of processing sensor data. This sensor data would be packaged into a serial bit-stream for wireless transmission to a base electronics package. The base unit would further process the sensor data for submission to a networked virtual environment and possible retransmission along with other virtual environment data back the user.

The wearable electronics could be a wearable computer such as ViA II PC from ViA Inc. [Ref. 89.] The processor and batteries of Flex PCs are configured as a waist belt that can be easily and comfortably worn. Data from MARG sensors and the RF position system could be routed to such a wearable PC. The role of the wearable PC would be to collect and process the data into a desirable state vector form, and wirelessly transmit the state vector to a VE station. The state vector may contain body position coordinates and limb orientation data as well as other forms of body posture representation.

The wearable PC and the SE base station would be linked through a wireless local area network (WLAN). One possible implementation of the WLAN is to use Lucent Technologies WaveLAN wireless products. The minimum implementation requires one WavePOINT wireless bridge and one WaveLAN PCMCIA card. Based on experimental results, the data rate of such a WLAN is about 1.4 Mbps at a range of 100 meters. The range can be up to 300 meters, but data rate will decrease to 100~500 Kbps, depending on the transmission environment. The maximum data rate requirement for the body suit is 162Kbps, assuming that the body suit has fifteen 9-axis MARG sensors sampled at100Hz, and each axis is sampled by a 12-bit AD converter. If sensor data is processed by embedded microprocessors or by the wearable computer and 15 unit quaternions are transmitted, the bandwidth requirement could be decreased to approximately 54Kbps.

Power will be supplied by rechargeable, hot-swappable batteries. By making batteries hot-swappable, endurance would not be a problem as long as each battery charge allows operation for a reasonable length of time.

F. FILTERING

The theory and development of the quaternion attitude filter described in this document and in [Ref. 51.] is largely complete. Experimental work involving the selection of filter gains in various operating regimes remains to be completed. Though Kalman filters are considered statistically optimal, it remains to be see whether such a filter could be developed for and would be of benefit to this application. Only after a Kalman filter has been developed will it be possible to determine whether it would be a better choice than the complementary filter based on Gauss-Newton iteration described in this research.

Kalman filtering is highly dependent on the quality of the incorporated process model. When applied to human body motion tracking, Kalman filter design requires an adequate dynamic model of the human musculoskeletal system, and the measurement statistics of the MARG sensors and RF positioning system to be used [Ref. 14.]. Dynamic models of the musculoskeletal system are well established and widely used for computer simulations of human body motions [Ref. 34.]. These models are given in the form of second order differential equations containing parameters representing body segment mass, center of mass, and moments of inertia. Though these models are ideal for computer simulations of human body motions, they are computationally too complex to work in a system requiring real-time tracking of multiple users wearing multiple sensors.

One possible approach to the modeling problem is to develop a model that is adequate but not overwhelmingly complex. Each limb segment could be considered independently of the others, or possibly motions of upper body segments could be considered independently of motions of lower body segments. This approach suggests that the process model needed for Kalman filtering may not need to make use of articulated body models, but could treat each limb segment as a single rigid body moving under the influence of forces produced by muscles and connective tissues [Ref. 96.][Ref. 28.]. The availability of reliable MARG sensors allows the gathering of statistical data needed to construct the model.

[Ref. 48.] describes the preliminary development of a reduced order Kalman filter for body tracking applications. To reduce the dimension of the state vector and simplify and linearize the state equations, Gauss-Newton iteration is utilized to compute the optimal quaternion relating measured to computed acceleration and magnetometer values. This filter work, like that described in [Ref. 27.], makes no attempt to model the dynamics of human motion.

G. A PROTOTYPE INERTIAL TRACKING BODY SUIT

In order to track a human in a VE, it will be necessary to outfit the user with a body suit. This suit would incorporate multiple MARG sensors, at least one RF position tracker and an electronics unit capable of processing sensor data. Avoiding encumbrance to the user and the method of sensor attachment would be primary concerns in designing the suit. Processing of data would be divided between the sensors themselves, the wearable PC and a base system. Decisions would have to be made regarding where exactly these divisions should be made. Such decisions would be driven by the need to reduce latency and increase resolution and registration. Factors involved would include transmission bandwidth and the processing power of the various components.

Two key factors must be considered when determining sensor placement and the method of attachment. The sensors must be reasonably stable relative to the bone structure of the user and the body suit and sensors must be easily donned. Relative motion between the bone structure and the sensor will be an additional source of noise and cause the sensors to report attitudes which do not correspond to the actual posture of the user. Most human models only attempt to approximate the human skeleton system. For instance the actual complexities of the shoulder and spine are not captured by the human model described in this research. Thus, sensors must be attached and placed in a manner which will not exaggerate the simplifications of the model. [Ref. 78.]

H. POSTURE DATA IN A NETWORKED SYNTHETIC ENVIRONMENT

Networked synthetic environments suffer from limitations of bandwidth, processing power and minimum transmission times. Work needs to be completed to facilitate the insertion of a high-resolution human into a networked synthetic environment. This goal requires research into different methods of encapsulating gesture data and the trade-offs involved in processing at various nodes in the network. Once this has been completed an efficient method for sending and processing this data could be developed.

Quaternion representation of orientation allows all attitudes to be represented without singularities. If a human model is composed of 15 separate segments, describing this posture using unit quaternions requires 45 floating point numbers. If the same model has 60 degrees of freedom, then 60 joint angles must be transmitted. Transmission of homogenous transform matrices will require five times the bandwidth of either method. Joint angle representation will require the use of forward kinematics. Update of the posture of a 15 segment human model using quaternions will require 840 scalar operations. This is an order of magnitude less than the 3,780 scalar operations needed to reset the posture using transform matrices or joint angles. Quaternions do not allow the possibility of applying joint constraints, but given adequate tracking accuracy this should not be a drawback.

The lag or delay of the posture data being received at remote nodes of the network presents another problem area. For instance, in a virtual battlefield simulation network delays may cause entities to be targeted based on a position they no longer occupy since an updated position has not yet been received. It is not likely that network transmission times will be significantly reduced in the near future. A common approach to this problem is prediction or dead reckoning based on the last update received [Ref. 98.]. Of course, predicting the future position or posture of a human is more difficult than predicting that of a vehicle due to numerous degrees of freedom, and the wide ranges of speed and motion in each limb segment. To accomplish such a prediction, velocity or rate data as well as location data must be sent across the network. Between position updates, the latest velocity data could be integrated or Kalman filtering could be used to predict the current position. Using this scheme, position updates could be sent only occasionally to correct errors due to inaccuracies in the velocity measurements.

I. CONCLUSIONS

This research has demonstrated a new technology for tracking the posture of an articulated rigid body. The technology is based on the use of inertial/magnetic sensors to independently determine the orientation of each link in the rigid body. Though the primary application described here was motion capture for inserting humans into networked virtual environments, inertial/magnetic orientation tracking could be applied to a broad range of problems which require tracking of an articulated structure without being continuously dependent upon an artificially generated source. The articulated body can be either animal or machine.

At the core of the system is an efficient complementary filter which uses a quaternion representation of orientation. Formulation of the filter is based upon the Orthogonal Quaternion Theorem which is presented and proved in this document. Error minimization is accomplished using Gauss-Newton iteration. The filter can continuously track the orientation of human body limb segments through in all attitudes without singularities. Drift corrections are made continuously with no requirement for still periods. Though the filter is nonlinear, it is shown through nonlinear simulations and actual system performance that linear analysis of the filter is relevant and can by used as a method for selecting scale factors and predicting performance.

The filter processes data from MARG sensors which contain components typically combined to form an inertial navigation system. The sensor has nine axes which include three orthogonal angular rate sensors, three orthogonal linear accelerometers and three orthogonal magnetometers. All sensor components are of a small form factor. Methods for conditioning and digitizing the output of the individual components are presented. Magnetometer and accelerometer data are used to create earth-fixed reference vectors. Rate sensor data is used to quicken the orientation estimates. While this quickening is typically necessary in feedback control applications, it may not be needed in low acceleration applications [Ref. 57.]. Sensor calibration is achieved using a novel calibration routine which requires no specialized equipment.

Articulated body posture is represented using a model based entirely on quaternion/vector pairs. Individual limb segments are oriented independently using a quaternion representation of the orientation relative to an earth-fixed reference frame. The model is mathematically simple. This simplicity reduces significantly the number of calculations needed to set the model posture. The underlying simplicity makes possible a quick and accurate calibration algorithm which compensates for misalignments between sensor and limb segment coordinate axes. The model may be adjusted to match the anthropometric measurements of an individual human subject.

The implemented system tracks human limb segments accurately with a 100 Hz update rate. Experimental results demonstrate that inertial/magnetic orientation estimation is a practical method of tracking human body posture. With additional sensors, the architecture produced could be easily scaled for full body tracking. This new technology overcomes the limitations of motion tracking technologies currently in use. It is potentially capable of providing wide area tracking of multiple users for synthetic environments and augmented reality applications.

APPENDIX A. DERIVATION OF GAUSS-NEWTON ITERATION EQUATIONS

Eq. (5.9) defines the quaternion filter error vector as

$$\vec{\varepsilon}(\hat{q}) = \vec{y}_0 - \vec{y}(\hat{q}) \tag{5.9}$$

where $\vec{y}_0$ denotes a measured value and $\vec{y}(\hat{q})$ is the calculated value based on the current estimate $\hat{q}$. The square of the error or the scalar squared error criterion function is given by Eq. (5.10)

$$\varphi(\hat{q}) = \vec{\varepsilon}^T(\hat{q})\vec{\varepsilon}(\hat{q}) \tag{5.10}$$

The criterion function is minimized by finding an "adjustment" to $\hat{q}$ termed Δq.

The non-linear function, $\vec{y}(\hat{q})$ can be approximated by linearizing about $\hat{q}$. The linearization is completed using the first two terms in the Taylor series expansion

$$\vec{y}(\hat{q} + \Delta q) = \vec{y}(\hat{q}) + X\Delta q + O(\Delta q^2) \tag{A.1}$$

where $\hat{q}$ and Δq are treated as four-space column vectors and X is the 6 x 4 multi-dimensional derivative of $y(\hat{q})$ with respective to $\hat{q}$. (See Appendix B for further discussion of the X matrix) Ignoring the non-linear portion of the Taylor series expansion and substituting Eq. (A.1) into Eq. (5.9) produces a linear approximation of the error vector.

$$\vec{\varepsilon}(\hat{q} + \Delta q) = y_0 - \vec{y}(\hat{q}) - X\Delta q = \vec{\varepsilon}(\hat{q}) - X\Delta q \tag{A.2}$$

From the inverse law of transposed products, it follows that

$$\vec{y}(\hat{q} + \Delta q)^T = \vec{\varepsilon}(\hat{q})^T - \Delta q^T X^T \tag{A.3}$$

Thus from Eq. (A.2) and Eq. (A.3), the criterion function can be approximated by

$$\varphi(\hat{q}) = \vec{\varepsilon}(\hat{q})^T\vec{\varepsilon}(\hat{q}) - \vec{\varepsilon}(\hat{q})^T X\Delta q - \Delta q^T X^T\vec{\varepsilon}(\hat{q}) + \Delta q^T X^T X\Delta q \tag{A.4}$$

If X is of full rank (full column rank) this is a positive definite form in Δq. Each of the terms in Eq. (A.4) evaluates to a scalar. By noting this fact and again using the inverse law of transposed matrices, it follows that

$$\vec{\varepsilon}(\hat{q})^T X\Delta q = \Delta q^T X^T \vec{\varepsilon}(\hat{q}) \tag{A.5}$$

This result allows the approximated error criterion function to be written

$$\varphi(\hat{q}) = \vec{\varepsilon}(\hat{q})^T \vec{\varepsilon}(\hat{q}) - 2\Delta q^T X^T \vec{\varepsilon}(\hat{q}) + \Delta q^T X^T X \Delta q \tag{A.6}$$

From differential calculus, the minimum or maximum of a function occurs where the slope of its tangent or derivative is equal to zero. The *gradient* (vector derivative) of squared error criterion function, $\varphi(\hat{q})$ is given by

$$\frac{d\varphi}{d\hat{q}} = -2X^T \vec{\varepsilon}(\hat{q}) + 2X^T X \Delta q \tag{A.7}$$

When the criterion function is a positive definite form, the unique minimum of Eq. (A.6) is found by equating the gradient to zero and solving for Δq. This result is the Gauss-Newton step given by Eq. (5.11) as

$$\Delta q = \left[X^T X\right]^{-1} X^T \varepsilon(\hat{q}) = S^{-1} X^T \varepsilon(\hat{q}) \tag{5.11}$$

The above approximation ignores the $O(\Delta q^2)$ term based on the assumption that Eq. (5.11) will be evaluated iteratively [Ref. 51.]. Simulations have demonstrated that Newton iteration, which takes in account this term, performs no better if it is assumed that all estimation errors are relatively small.

APPENDIX B. DERIVATION OF THE X MATRIX

There are an infinite number of quaternions that can be used to represent any given orientation. These quaternions differ by a scalar multiplier. If q is a unit quaternion and $\tilde{q} = \alpha q$ where α is any non-zero scalar, then

$$|q|^2 = \alpha^2 \tag{B.1}$$

and

$$\tilde{q}v\tilde{q}^{-1} = \alpha q v\frac{\alpha\bar{q}}{\alpha^2} = qvq^{-1} \tag{B.2}$$

The elements of the 6 x 4 X matrix are the partial derivatives of the computed measurement vector, $\vec{y}(\hat{q})$ with respect to each of the components of the estimated orientation quaternion, $\hat{q}$.

Given

$$\vec{y}(\hat{q}) = [Ve(\hat{q}^{-1}m\hat{q}), Ve(\hat{q}^{-1}n\hat{q})]^T = [\hat{h}_1\hat{h}_2\hat{h}_3\hat{b}_1\hat{b}_2\hat{b}_3]^T \tag{B.3}$$

Then, the X matrix is given by

$$X = \begin{bmatrix} \frac{\partial\hat{h}_1}{\partial\hat{q}_0} & \frac{\partial\hat{h}_1}{\partial\hat{q}_1} & \frac{\partial\hat{h}_1}{\partial\hat{q}_2} & \frac{\partial\hat{h}_1}{\partial\hat{q}_3} \\ \frac{\partial\hat{h}_2}{\partial\hat{q}_0} & \frac{\partial\hat{h}_2}{\partial\hat{q}_1} & \frac{\partial\hat{h}_2}{\partial\hat{q}_2} & \frac{\partial\hat{h}_2}{\partial\hat{q}_3} \\ \frac{\partial\hat{h}_3}{\partial\hat{q}_0} & \frac{\partial\hat{h}_3}{\partial\hat{q}_1} & \frac{\partial\hat{h}_3}{\partial\hat{q}_2} & \frac{\partial\hat{h}_3}{\partial\hat{q}_3} \\ \frac{\partial\hat{b}_1}{\partial\hat{q}_0} & \frac{\partial\hat{b}_1}{\partial\hat{q}_1} & \frac{\partial\hat{b}_1}{\partial\hat{q}_2} & \frac{\partial\hat{b}_1}{\partial\hat{q}_3} \\ \frac{\partial\hat{b}_2}{\partial\hat{q}_0} & \frac{\partial\hat{b}_2}{\partial\hat{q}_1} & \frac{\partial\hat{b}_2}{\partial\hat{q}_2} & \frac{\partial\hat{b}_2}{\partial\hat{q}_3} \\ \frac{\partial\hat{b}_3}{\partial\hat{q}_0} & \frac{\partial\hat{b}_3}{\partial\hat{q}_1} & \frac{\partial\hat{b}_3}{\partial\hat{q}_2} & \frac{\partial\hat{b}_3}{\partial\hat{q}_3} \end{bmatrix} \tag{B.4}$$

The ith column of X is

$$\frac{\partial\vec{y}}{\partial\hat{q}_i} = \frac{\partial}{\partial\hat{q}_i}(\hat{q}^{-1}m\hat{q}, \hat{q}^{-1}n\hat{q}) \tag{B.5}$$

By the product rule of differential calculus, it follows that

$$\frac{\partial \vec{y}}{\partial \hat{q}_0} = \left(\frac{\partial \hat{q}^{-1}}{\partial \hat{q}_0} m \hat{q} + \hat{q}^{-1} m \frac{\partial \hat{q}}{\partial \hat{q}_0},\ \frac{\partial \hat{q}^{-1}}{\partial \hat{q}_0} n \hat{q} + \hat{q}^{-1} n \frac{\partial \hat{q}}{\partial \hat{q}_0}\right) \tag{B.6}$$

Similarly for columns two through three

$$\frac{\partial \vec{y}}{\partial \hat{q}_1} = \left(\frac{\partial \hat{q}^{-1}}{\partial \hat{q}_1} m \hat{q} + \hat{q}^{-1} m \frac{\partial \hat{q}}{\partial \hat{q}_1}, \frac{\partial \hat{q}^{-1}}{\partial \hat{q}_1} n \hat{q} + \hat{q}^{-1} n \frac{\partial \hat{q}}{\partial \hat{q}_1}\right) \tag{B.7}$$

$$\frac{\partial \vec{y}}{\partial \hat{q}_2} = \left(\frac{\partial \hat{q}^{-1}}{\partial \hat{q}_2} m \hat{q} + \hat{q}^{-1} m \frac{\partial \hat{q}}{\partial \hat{q}_2}, \frac{\partial \hat{q}^{-1}}{\partial \hat{q}_2} n \hat{q} + \hat{q}^{-1} n \frac{\partial \hat{q}}{\partial \hat{q}_2}\right) \tag{B.8}$$

$$\frac{\partial \vec{y}}{\partial \hat{q}_3} = \left(\frac{\partial \hat{q}^{-1}}{\partial \hat{q}_3} m \hat{q} + \hat{q}^{-1} m \frac{\partial \hat{q}}{\partial \hat{q}_3}, \frac{\partial \hat{q}^{-1}}{\partial \hat{q}_3} n \hat{q} + \hat{q}^{-1} n \frac{\partial \hat{q}}{\partial \hat{q}_3}\right) \tag{B.9}$$

Also from the product rule of differential calculus

$$\frac{\partial}{\partial q_i}(q q^{-1}) = \frac{\partial q}{\partial q_i} q^{-1} + q \frac{\partial q^{-1}}{\partial q_i} = 0 \tag{B.10}$$

Solving for $\frac{\partial q^{-1}}{\partial q_i}$ produces

$$\frac{\partial q^{-1}}{\partial q_i} = -q^{-1} \frac{\partial q}{\partial q_i} q^{-1} \tag{B.11}$$

This result can be substituted into Eq. (B.6) through Eq. (B.9) to produce the general form

$$\frac{\partial \vec{y}}{\partial \hat{q}_i} = \left(\left(-q^{-1} \frac{\partial q}{\partial q_i} q^{-1}\right) m \hat{q} + \hat{q}^{-1} m \frac{\partial \hat{q}}{\partial \hat{q}_i},\ \left(-q^{-1} \frac{\partial q}{\partial q_i} q^{-1}\right) n \hat{q} + \hat{q}^{-1} n \frac{\partial \hat{q}}{\partial \hat{q}_i}\right) \tag{B.12}$$

To complete this derivation of X, the partial derivatives of $\vec{y}(\hat{q})$ with respect to a $\hat{q}$ of any length are written

$$\frac{\partial q}{\partial q_0} = (1\ 0\ 0\ 0) = 1 \tag{B.13}$$

$$\frac{\partial q}{\partial q_1} = (0\ 1\ 0\ 0) = i \tag{B.14}$$

$$\frac{\partial q}{\partial q_2} = (0\ 0\ 1\ 0) = j \tag{B.15}$$

$$\frac{\partial q}{\partial q_3} = (0\ 0\ 0\ 1) = k \tag{B.16}$$

When using a $\hat{q}$ of any arbitrary length, an X matrix derived using the above method will not be of full rank and the resulting regression matrix will be singular and non-invertible. This is due to the result given by Eq. (B.2). Constraining $\hat{q}$ to be of unit length will eliminate this problem.

The following partial derivatives of the inverse of q are derived by assuming q is a unit quaternion so that $q^{-1} = \bar{q}$. Under this assumption

$$\frac{\partial q^{-1}}{\partial q_0} = (1\ 0\ 0\ 0) = 1 \tag{B.17}$$

$$\frac{\partial q^{-1}}{\partial q_1} = (0\ -1\ 0\ 0) = -i \tag{B.18}$$

$$\frac{\partial q^{-1}}{\partial q_2} = (0\ 0\ -1\ 0) = -j \tag{B.19}$$

$$\frac{\partial q^{-1}}{\partial q_3} = (0\ 0\ 0\ -1) = -k \tag{B.20}$$

Since Eq. (B.17) through Eq. (B.20) are partial derivatives of constrained vectors, Lagrange multipliers should be used in their formulation [Ref. 76.]. Evidently, Lagrange multipliers were not used in the simple derivations shown here. However, computational experiments show that the X matrix so derived is of full rank so that the inverse of the regression matrix $S = X^T X$ exists and can be used to correctly obtain $\hat{q}$ by Gauss-Newton iteration [Ref. 51.]. In this dissertation, this 4 x 4 problem is further reduced by combining Eq. (B.17) through Eq. (B.20) with the orthogonal quaternion theorem to achieve a still simpler 3 x 3 matrix inversion Gauss-Newton method.

APPENDIX C. VIDEO DEMONSTRATION

This section briefly describes the contents of each portion the video appendix.

1. Sensor Calibration

This portion demonstrates the sensor calibration algorithm. The calibration process is described as each step is completed. For comparison purposes, tracking performance is shown both before and after calibration.

2. Body Model Calibration

This portion demonstrates the body model calibration algorithm. The reference position is described and visually displayed. The effect of the calibration algorithm on the displayed posture can be seen.

3. Posture Tracking

This portion demonstrates the dynamic performance of the prototype inertial/ magnetic body tracking system. The tracking of various limb segments is shown. Adjustment of the model dimensions is performed to allow display of closed loop postures. Various filter gains are used through out this video segment.

4. Magnetic/Gravity Tracking

Dynamic performance of the system without the use of rate sensor data is shown.

5. Reduced Drift Correction

The dynamic performance of the system when performing drift correction at a rate of approximately 6 Hz is shown. Overall performance is observed to vary little from that seen in the Posture Tracking segment of the video demonstration.

6. InterSense InertiaCube

Video demonstration of the inability of the filtering algorithms associated with the InterSense InertiaCube to continuously correct for drift. This performance is contrasted visually against the MARG sensor and the quaternion filter algorithm.

LIST OF REFERENCES

1. Adelstein, B., Johnston, E., and Ellis S., "Dynamic Response of Electromagnetic Spatial Displacement Trackers," *Presence: Teleoperators and Virtual Environments*, Vol. 5, No. 3, MIT Press, Cambridge, MA, Summer 1996, pp. 302-318.

2. Analog Devices Inc., *ADXL202/210 Data Sheet*, http://www.analog.com/pdf/ADXL202_10_b.pdf, 2000.

3. Analog Devices Inc., *ADXL05 Data Sheet*, Analog Devices, Inc., Norwood, MA, 1996.

4. Ascension Technology Corporation, "Overcoming Metal Problems: The Advantage of DC Magnetic Tracking," http://www.ascension-tech.com/oldsite/dcadvantage.htm, 1997.

5. Azuma, R., *Predictive Tracking for Augmented Reality*, Ph.D. Dissertation, University of North Carolina at Chapel Hill, Computer Science Technical Report TR#95-007, Chapel Hill, NC, February 1995.

6. Bachmann, E., Duman I., Usta, U., McGhee R., Yun, X., and Zyda, M., "Orientation tracking for Humans and Robots Using Inertial Sensors," *International Symposium on Computational Intelligence in Robotics & Automation* (CIRA 99), Monterey, CA, November 1999, pp.187-194.

7. Bachmann, E., McGhee, R., Whalen, R., Steven, R., Walker, R., Clynch, J., Healey, A., and Yun, X., "Evaluation of an Integrated GPS/INS System for Shallow-water AUV Navigation (SANS)," *Proceedings of the IEEE Symposium on Autonomous Underwater Vehicle Technology*, AUV '96, Monterey, CA, 3-6 June, 1996, pp. 268-275.

8. Bachmann, E., *Research Notes: Quaternion Attitude Filter*, Computer Science Department, Naval Postgraduate School, Monterey, CA, 1996

9. Banks, D., *Introduction to Microengineering: MEMS, Micromachines, MST*, http://www.dbanks.demon.co.uk/ueng/index.html, 1999.

10. Badler, N., Hollick, M., and Granieri, J., "Real-Time Control of a Virtual Human Using Minimal Sensors," *Presence: Teleoperators and Virtual Environments*, Vol. 2, No. 1, MIT Press, Cambridge MA, 1993, pp. 82-86.

11. Badler, N., Phillips, C., and Webber, B., *Simulating Humans: Computer Graphics Animation and Control*, Oxford University Press, Inc., New York, NY, 1993.

12. Bers J., "A Model Server for Human Motion Capture and Representation," *Presence: Teleoperators and Virtual Environments*, Vol. 5, No. 4, MIT Press, Cambridge MA, Fall 1996, pp. 381-392.

13. Bible, S., Zyda, M., and Brutzman, D., "Using Spread-Spectrum Ranging Techniques for Position Tracking in a Virtual Environment," *Proceedings of the 1995 Workshop on Networked Realities*, Boston, MA, October 26-28, 1995.

14. Brown, R., Hwang, P., *Introduction to Random Signals and Applied Kalman Filtering, Second Edition*, John Wiley and Sons, Inc., New York, NY, 1992.

15. Caruso, M., *Set/Reset Pulse Circuits for Magnetic Sensors*, Honeywell, Inc., AN-201, 1995.

16. Cooke, J., Zyda, M., Pratt, D., McGhee, R., "NPSNET: Flight Simulation Modeling Using Quaternions," *Presence*, Vol. 1, No. 4, MIT Press, Cambridge MA, Fall, 1992, pp. 404-420.

17. Craig, J., *Introduction to Robotics: Mechanics and Control, Second Edition,* Addison-Wesley Publishing company, Inc., Menlo Park, CA, 1989.

18. *Crossbow CXL04M3 Data Sheet*, Crossbow Inc., 1998.

19. Cyberware Inc., http://www.cyberware.com/, 2000.

20. Duman, I., *Design, Implementation and Testing of a Real-Time Software System for a Quaternion-Based Attitude Estimation Filter,* Master's Thesis, Naval Postgraduate School, Monterey, CA, March 1999.

21. Durlach, N., and Mavor, A., National Research Council, *Virtual Reality: Scientific and Technological Challenges*, National Academy Press, Washington, DC, 1995.

22. FakeSpace Inc., *Boom 3C*, http://www.fakespace.com/products/boom3c.html.

23. Featherstone, R., and Orin, D., "Robot Dynamics: Equations and Algorithms," *Proceedings of the 2000 IEEE International Conference on Robotics and Automation*, San Francisco, CA, 2000.

24. Feng, J., *An RF Head-Tracker*, Phase I Final Report, NAVAIR SBIR Topic No. N95-144, October 1996.

25. Ferrin, F., "Survey of Helmet Tracking Technologies," *SPIE Proceedings, Large-Screen-Projection, Avionic, and Helmet-Mounted Displays*, 1991.

26. Foxlin, E., "Constellation: A Wide-Range Wireless Motion-Tracking System for Augmented Reality and Virtual Set Applications," *Proceedings of SIGGRAPH '98*, ACM, 1998, pp. 371-378.

27. Foxlin, E., "Inertial Head-Tracker Sensor Fusion by a Complementary Separate-Bias Kalman Filter," *Proceedings of VRAIS '96*, IEEE, 1996, pp. 185-194.

28. Frey, W., *Application of Inertial Sensors and Flux-Gate Magnetometers to Real-Time Human Body Motion Capture*, Master's Thesis, Naval Postgraduate School, Monterey, CA, September 1996.

29. Fuchs, E., *Inertial Head-Tracking*, Master's Thesis, Massachusetts Institute of Technology, Cambridge MA, 1993.

30. Funda, J., Taylor, R., and Paul, R., "On Homogeneous Transforms, Quaternions, and Computational Efficiency," *IEEE Transactions on Robotics and Automation*, Vol. 6, No. 3, June 1990, pp. 382-388.

31. Haas Automation, Inc., *Haas Model: TRT-7 Tilting Rotary Table*, Chatsworth, CA, July 1992.

32. Motion Analysis Corporation, *HiRES 3D Motion Capture System*, http://www.motionanalysis.com/index.html, Santa Rosa, CA, 2000

33. Humanoid Animation Working Group, *Specification for a Standard Humanoid, Version 1.1*, http://h-anim.org/spec1.1/, 1999.

34. *Human Motion Analysis, Current Applications and Future Directions*, ed. by G. F. Harris and P. A. Smith, IEEE Press, Piscataway, New Jersey, 1996.

35. Henault, G., *A Computer Simulation Study and Component Evaluation for a Quaternion Filter for Sourceless Tracking of Human Limb Segment Motion*, Master's Thesis, Naval Postgraduate School, Monterey, CA, March 1997.

36. InterSense, Inc., http://www.isense.com, Internet, 1997.

37. InterSense, Inc., *IS-300 Series Users Manual*, Cambridge MA, 1997.

38. *Honeywell Three-Axis Magnetoresistive Sensor, HMC1023*, Honeywell Inc., 1999.

39. *Honeywell Three-Axis Magnetic Sensor Hybrid, HMC2003*, Honeywell Inc., 1998.

40. Jones, L., "Dextrous Hands: Human, Prosthetic, and Robotic," *Presence: Teleoperators and Virtual Environments*, Vol. 6, No. 1, February 1997, pp. 29-56.

41. Kuo, E., *Automatic Control Systems, Seventh Edition*, Prentice Hall, Inc., Englewood Cliffs, NJ, 1995.

42. Kuipers, J., *Quaternions and Rotation Sequences*, Princeton University Press, Inc., Princeton, NJ, 1998.

43. Lawrence, A., *Modern Inertial Technology: Navigation, Guidance, and Control, Second Edition*, Springer-Verlag Inc., New York, NY, 1998.

44. Livingston, M. and State A., "Magnetic Tracker Calibration for Improved Augmented Reality Registration", *Presence: Teleoperators and Virtual Environments*, Vol. 6, No. 5, 1997, pp. 532-546.

45. Logston, T., *The Navstar Global Positioning System*, Van Nostrand Reinhold, New York, NY, 1992.

46. Nixon, M., McCallum, M., Fright, W., Price, N., "The Effects of Metals and Interfering Fields on Electromagnetic Trackers," *Presence: Teleoperators and Virtual Environments*, Vol. 7, No. 2, MIT Press, Cambridge, MA, 1998, pp. 204-218.

47. MacKenzie, S., "Input Devices and Interaction Techniques for Advanced Computing," *Virtual Environments and Advanced Interface Design*, Oxford University Press, Inc., New York, NY, 1995.

48. Marins, J., *An Extended Kalman Filter for Quaternion-Based Attitude Estimation,* Engineer's Thesis, Naval Postgraduate School, Monterey, CA, September 2000.

49. Maybeck, P., "The Kalman Filter: An Introduction to Concepts," Tutorial Presented at the *IEEE Oceanic Engineering Society Symposium on Autonomous Underwater Vehicle Technology*, Monterey, CA, 1996.

50. McGhee, R., *Research Notes: Crossover Frequency*, Computer Science Department, Naval Postgraduate School, Monterey, CA, 2000.

51. McGhee, R., Bachmann, E., Yun X. and Zyda, M., "Real-Time Tracking and Display of Human Limb Segment Motions Using Sourceless Sensors and a Quaternion-Based Filtering Algorithm - Part I: Theory," MOVES Academic Group Technical Report NPS-MV-01-001, Naval Postgraduate School, Monterey, CA, November 2000.

52. McGhee, R., Bachmann, E., Brutzman, D., and Zyda, M., "Rigid Body Dynamics, Inertial Reference Frames, and Graphics Coordinate Systems: A Resolution of Conflicting Conventions and Terminology," MOVES Academic Group Technical Report NPS-MV-01-002, Naval Postgraduate School, Monterey, CA, November 2000.

53. McGhee, R., *CS3472 Class Notes: Quaternions Demystified*, Computer Science Department, Naval Postgraduate School, Monterey, CA, 1999.

54. McGhee, R., *Research Notes: Linearization of a Quaternion Attitude Filter*, Computer Science Department, Naval Postgraduate School, Monterey, CA, 1998.

55. McGhee, R., *Research Notes: Response of Quaternion Filter to Initial Condition Errors*, Computer Science Department, Naval Postgraduate School, Monterey, CA, 1998.

56. McGhee, R., *CS3472 Class Notes: Transfer Functions and Time Response from Block Diagrams*, Computer Science Department, Naval Postgraduate School, Monterey, CA, 1996.

57. McGhee, R., Nakano, E., Koyachi, N., and Acachi, H., "An Approach to Computer Coordination of Motion for Energy-Efficient Walking Machines," *Bulletin of Mechanical Engineering Laboratory, Japan*, No. 43, 1986.

58. McGhee, R., and Walford, R., "A Monte Carlo Approach to the Evaluation of Conditional Expectation Parameter Estimates for Nonlinear Dynamic Systems," *IEEE Transactions on Automatic Control*, Vol. AC-13, No. 1, February 1968, pp. 29-37.

59. McGhee, R., "Some Parameter-Optimization Techniques," *Digital Computer User's Handbook*, McGraw-Hill, 1967, pp. 234-253.

60. McGhee, R., *Identification of Nonlinear Dynamic Systems by Regression Analysis Methods*, Ph.D. Dissertation, University of Southern California, 1963.

61. McKinney Technology, Doug Mckinney, 9 Glen Avenue, Prunedale, CA, 93907.

62. Meyer, K., Applewhite, H., and Biocca, F., "A Survey of Position Trackers," *Presence: Teleoperators and Virtual Environments*, Vol. 1, No. 2, Spring 1992, pp. 173-200.

63. *MicroSensors, Advance Information, Silicon MicoRing Gyro*, MicroSensors, Inc., Costa Mesa, CA, 1998.

64. Molet, T., Boulic R., and Thalmann D, "Human Motion Capture Driven by Orientation Measurements," *Presence: Teleoperators and Virtual Environments*, Vol. 8, No. 2, MIT Press, Cambridge MA, 1999, pp. 187-203.

65. Molet, T., Aubel, A., Tolga, C., Carion, S., Lee, E., Naagnenat-Thalmann, N., Hoser, H., Pandzic, I., Sannier, G., and Thalmann, D., "Anyone for Tennis," *Presence: Teleoperators and Virtual Environments*, Vol. 8, No. 2, MIT Press, Cambridge MA, 1999, pp. 140-156.

66. Morris, T., and Donath M., "Using a Maximum Error Statistic to Evaluate Measurement Errors in 3D Position and Orientation Tracking Systems", *Presence: Teleoperators and Virtual Environments*, Vol. 2, No. 4, MIT Press, Cambridge MA, Fall 1993, pp. 314-343.

67. Mudler, A., "Human Movement Tracking Technology: Resources," Addendum to Technical Report 94-1, School of Kinesiology, Simon Fraser University, July 1994.

68. National Instruments Corporation, http://www.natinst.com, 1998.

69. National Instruments Corporation, DAQ, PCI E Series User Manual, Part Number 320945C-01, July 1997.

70. Peak Motus Motion Measurement System, http://www.peakperform.com/, Peak Performance Technologies, Inc., 2000.

71. Polhemus Inc., "STAR TRAK Real-time RF Wireless Motion Capture System," http://www.polhemus.com/stards.htm, 2000.

72. Srikanth S., and Reddy N., "EMG-Based Interface for Position Tracking and Control in VR Environments and Teleoperation," *Presence: Teleoperators and Virtual Environments*, Vol. 6, No. 3, MIT Press, Cambridge MA, 1997, pp. 282-291.

73. Raab, F., Blood, E., Steiner, O., and Jones, H., "Magnetic Position and Orientation Tracking System," *IEEE Transactions on Aerospace and Electronics Systems*, AES-15, No. 5, 1979, pp. 709-717.

74. Reality Fusion Inc., "GameCam," http://www.realityfusion.com/, 2000.

75. Roberts, R., *Analysis, Experimental Evaluation, and Software Upgrade for Attitude Estimation by the Shallow-water AUV Navigation System (SANS)*, Master's Thesis, Naval Postgraduate School, Monterey, CA, March 1997.

76. Rust, B., and Burrus, W., *Mathematical Programming and the Numerical Solution of Linear Equations,* American Elsevier Publishing company, Inc., New York, NY, 1972.

77. Sakaguchi, T., Kanamori, T., Katayose, H., Sato, K., and Inokuchi, S., "Human Motion Capture by Integrating Gyroscopes and Accelerometers," *IEEE/SICE/RSJ International Conference on Multisensor fusion and Integration for Intelligent Systems*, December 1996. pp. 470-475.

78. Skopowski, P., *Immersive Articulation of the Human Upper Body in a Virtual Environment*, Master's Thesis, Naval Postgraduate School, Monterey, CA, December 1996.

79. Sudhanshu K., Hightower, R., and Stansfield, S., "Mapping Algorithms for Real-Time Control of an Avatar Using Eight Sensors," *Presence: Teleoperators and Virtual Environments*, Vol. 7, No. 1, MIT Press, Cambridge MA, 1998, pp. 1-21.

80. Sutherland, I. "A Head-mounted Three-dimensional Display," *Fall Joint Computer Conference, AFIPS Conference Proceedings*, 1968, pp. 757- 764.

81. *TLC2543C, TLC2543I, TLC2543M 12-Bit Analog-to-Digital Converters with Serial Control and 11 Analog Inputs*, Texas Instruments, Inc., http://www-s.ti.com/sc/psheets/slas079e/slas079e.pdf, 1993, Revised 2000.

82. *Tokin MDP-A3U7 3D Motion Sensor Data Sheet*, Tokin American Inc., http://www.tokin.com/Tokin_America_Products/46/p46a/p46a.html, 2000.

83. *Tokin CG-L33 Solid State Rate Sensor Data Sheet*, Tokin American Inc., http://www.tokin.com/Tokin_America_Products/46/p46c/p46c.html, 2000.

84. *Tokin CG16D Solid State Rate Sensor Data Sheet*, Tokin American Inc., http://www.tokin.com/Tokin_America_Products/46/p46c/p46c.html, 1998.

85. Tolani, D., and Badler, N., "Real-Time Inverse Kinematics of the Human Arm," *Presence: Teleoperators and Virtual Environments*, Vol. 5, No. 4, MIT Press, Cambridge MA, 1996, pp. 393-401

86. Truxal, J., *Automatic Feedback Control System Synthesis*, McGraw-Hill Book Company, Inc., New York, NY, 1955.

87. Sabel, J., "Optical 3D Motion Measurement," *IEEE Instrumentation and Measurement Technology Conference*, Vol. 1, 1996, pp. 367-370.

88. Usta, U., *Comparison of Quaternion and Euler Angle Methods for Joint Angle Animation of Human Figure Models*, Master's Thesis, Naval Postgraduate School, Monterey, CA, March 1999.

89. Via II PC, http://www.flexipc.com/, Via Inc., 1999.

90. Vicon Motion Systems, http://www.metrics.co.uk/, Oxford Metric Ltd., Oxford, England.

91. Washburn, A., "A Short Introduction to Kalman Filters," Class Notes, Naval Postgraduate School, Monterey, CA, 1995.

92. Watt, A., and Watt, M., *Advanced Animation and Rendering Techniques, Theory and Practice*, ACM Press, New York, NY, 1992.

93. Welch, G., Bishop, G., Vicci, L., Brumback, S., Keller, L., Colucci, D., "The HiBall Tracker: High-Performance Wide-Area Tracking for Virtual and Augmented Environments," *Proceedings of the ACM Symposium on Virtual Reality Software and Technology 1999 (VRST 99)*, University College London, 1999.

94. Welch, G., *SCAAT: Incremental Tracking with Incomplete Information*, Ph.D. Dissertation, University of North Carolina at Chapel Hill, Computer Science Technical Report, TR 96-051, Chapel Hill, NC, 1996.

95. Wloka, M., "Lag in Multiprocessor Virtual Reality," *Presence: Teleoperators and Virtual Environments*, Vol. 4, No. 1, MIT Press, Cambridge MA, Winter 1995, pp. 50-63.

96. Yun, X., Bachmann, E., McGhee R., Whalen, R., Roberts, R., Knapp, R., Healey, A., and Zyda M.,"Testing and Evaluation of an Integrated GPS/INS System for Small AUV Navigation," *IEEE Journal of Ocean Engineering*, Vol. 24, No. 3, July 1999. pp. 396-404.

97. Zyda M., National Research Council, *Modeling and Simulation, Linking Entertainment and Defense*, National Academy Press, Washington, DC, 1997.

98. Zyda M., and Singhal, S., *Networked Virtual Environments, Design and Implementation*, ACM Press, New York, NY, 1999.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center. 2
8725 John J. Kingman Road., Ste 0944
Ft. Belvoir, VA 22060-6218

2. Dudley Knox Library 2
Naval Postgraduate School
411 Dyer Rd.
Monterey, CA 93943-5101

3. Steven Arms 2
President, MicroStrain, Inc.
294 N. Winooski Ave.
Burlington, VT 05401

4. Eric Bachmann, Code CS/Bc 10
Department of Computer Science
Naval Postgraduate School
Monterey, CA 93943-5118

5. Dr. Dan Boger, Code CS/Bo. 1
Department of Computer Science
Naval Postgraduate School
Monterey, CA 93943-5118

6. Dr.Don Brutzman, Code UW/Br 1
Undersea Warfare Academic Group
Naval Postgraduate School
Monterey, CA 93943-5118

7. Capt. Richard Bump 1
CNO, N6M
2000 Navy Pentagon
Washington, DC 20350-2000

8. Dr. Michael Coyle. 3
Program Manager, Engineering Sciences Directorate,
Mathematical and Computer Sciences Division
U.S. Army Research Office
P.O. Box 12211, 4300 S. Miami Blvd.
Research Triangle Park, NC 27709

9. Dr. James Clynch, Code OC/Cl 1
Department of Oceanography
Naval Postgraduate School
Monterey, CA 93943-5118

10. Dr. Ann Flood 1
1010 Cass Street, Suite D-10
Monterey, CA 93940

11. Dr. Anthony Healey, Code ME/Hy 1
Department of Mechanical Engineering
Naval Postgraduate School
Monterey, CA 93943-5118

12. Dr. Keebom Kang, Code SM/. 1
Department of Systems Management
Naval Postgraduate School
Monterey, CA 93943-5118

13. Dr. Ming Lin 1
University of North Carolina
Dept. of Computer Science
209 Sitterson Hall
Chapel Hill, NC 27599-3175

14. Dr. Lucia Luqi, Code CS/Lq. 1
Department of Computer Science
Naval Postgraduate School
Monterey, CA 93943-5118

15. Dr. Robert McGhee, Code CS/Mz 1
Department of Computer Science
Naval Postgraduate School
Monterey, CA 93943-5118

16. Doug McKinney 1
President, McKinney Technologies
9 Glen Avenue
Prundale, CA 93907

17. George Phillips 3
CNO, N6M1
2000 Navy Pentagon
Washington, DC 20350-2000

18. Dr. Man-Tak Shing, Code CS/Sh 1
Department of Computer Science
Naval Postgraduate School
Monterey, CA 93943-5118

19. Dr. Douglas Troy 1
Chair, Department of Computer Science & Systems Analysis
230-K Kreger
Miami University
Oxford, OH 45056

20. Dr. Greg Welch 2
University of North Carolina
Department of Computer Science
209 Sitterson Hall
Chapel Hill, NC 27599-3175

21. Russ Whalen 1
68-104 Au Street
Waialua, HI 96791-9438

22. Dr. Xiaoping Yun, Code EE/Yu 1
Department of Electrical and Computer Engineering
Naval Postgraduate School
Monterey, CA 93943-5118

23. Dr. Michael Zyda, Code MOVES/mz 1
Chair, Modeling, Virtual Environments & Simulation (MOVES) Academic Group
Naval Postgraduate School
Monterey, California 93943-5118